21

UCRL- 92909
PREPRINT

ICARUS

Steven B. Sutton

This paper was prepared for submittal to
Micro-CAE Technology Transfer Workshop
Flagstaff, AZ - August 18-23, 1985

July 1985

Lawrence
Livermore
National
Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

TF 85-12

The ICARUS Family of Computer Programs for Solving One-Dimensional Heat Conduction Problems on the IBM Personal Computer

Steven B. Sutton
Thermo-Fluid Mechanics Group
Mechanical Engineering Department
Lawrence Livermore National Laboratory

July 11, 1985

ABSTRACT

A family of computer programs for calculating one-dimensional planar, cylindrical or spherical conduction heat transfer are described. The programs include an interpretive BASIC heat conduction program, a compiled FORTRAN heat conduction program (with an interpretive BASIC pre-processor) and interpretive BASIC and compiled BASIC post-processing plot programs that work with both the BASIC and FORTRAN heat conduction programs.

The heat transfer programs can account for multiple material regions and complex boundary conditions (such as convection or radiation). In addition, the FORTRAN version of the program allows temperature dependent material properties and time or temperature dependent boundary conditions. Finite difference techniques are used to discretize the control volume energy balance equation. The resulting system of simultaneous algebraic equations are solved using a standard tri-diagonal reduction method. THe equations are formulated so that the solution can be fully implicit or fully explicit.

The BASIC plot post-processors allow the user to generate plots of the temperature or heat flux profile at specified times, or plots of the variation of temperature or heat flux with time at selected nodes. The programs are interactive for ease in use.

Several sample problems that compare numerical predictions to analytical solutions and other numerical predictions are discussed.

CONTENTS

1.0 INTRODUCTION

The family of micro-computer programs described in this document are an outgrowth of the ICARUS [1] heat conduction program that operates on the main-frame computers at the Lawrence Livermore National Laboratory. ICARUS had its origins in the mid-1970's when it was recognized that a one-dimensional heat conduction capability was needed for those problems where the geometry and boundary conditions made such dimensional simplification practical. Since its inception, the program has undergone extensive modification and improvement, and has been applied to problems dealing with laser fusion target fabrication, heat loads on underground tests, magnetic fusion switching tube anodes and nuclear waste isolation canisters.

To allow the program to be used by a wider community of users, a family of computer programs that operate on the IBM Personal Computer was derived from the main-frame computer program. The computer program family consists of:

ICARUSB - an interpretive BASIC heat conduction program. The program contains an interactive pre-processor block for collecting input information and the main calculation block for solving the heat conduction equation.

ICARUSF - a compiled FORTRAN heat conduction program. The program requires use of the separate interpretive BASIC pre-processor PREICAR to generate the input file for use in execution.

PREICAR - a compiled BASIC pre-processor for the FORTRAN heat conduction program ICARUSF. This routine is used for initial generation and modification of previously generated files.

PLOTIC - an interpretive BASIC plot post-processor for both the interpretive BASIC heat conduction program ICARUSB and the compiled FORTRAN heat conduction program ICARUSF.

CPLOTIC - a compiled BASIC plot post-processor for both the interpretive BASIC heat conduction program ICARUSB and the compiled FORTRAN heat conduction program ICARUSF. CPLOTIC differs from PLOTIC only in file directory capability and usage of the assembly language screen dump routine.

Each of the computer programs is described in detail in subsequent sections of this report.

The approximate computer core required to contain each of the programs is given below:

Program	Required Core (K bytes)
ICARUSB	64
ICARUSF	140
PREICAR	125
PLOTIC	95
CPLOTIC	95

Several comments on application of the heat transfer codes and "pit-falls" to avoid are given in Appendix A. It is suggested that the user become familiar with these suggestions before proceeding with using the codes.

2.0 AVAILABILITY

The following are made available:

- The sources for the five computer programs:
  1. The BASIC source for ICARUSB, consisting of 12 individual files.
  2. The FORTRAN source for ICARUSF, consisting of 16 individual files.
  3. The BASIC source for PREICAR consisting of one file.
  4. The BASIC source for PLOTIC consisting of one file.
  5. The BASIC source for CPLOTIC consisting of one file

- Compiled and loaded executables for the three programs:
  1. The FORTRAN heat conduction program ICARUSF [file ICARUSF.EXE].
  2. The BASIC program PREICAR [file PREICAR.EXE] which is a pre-processor for the FORTRAN program ICARUSF.
  3. The BASIC plot post-processing program CPLOTIC [file CPLOTIC.EXE].

The files may be obtained from the author by stating the programs desired, and providing formatted 5.25 inch double-sided, double-density floppy diskettes. Three floppy diskettes are required to obtain the entire package including sources and compiled executables.

3.0 GENERAL TERMINOLOGY

Throughout the descriptions of the individual computer programs, several terms will frequently be used. A familiarity with their meaning will improve the readability of those sections. The important terms are:

material block / geometry region - This is a region of the problem that is adjoined by other such regions. Each region may have a different material, or a different geometric zoning.

zoning - The division of the problem space into discrete segments and nodal points upon which finite differencing is performed.

gap - A "void" separating two material blocks or geometric regions. Radiative transport and convective transport may take place across the void.

non-linear iteration - In the computer program ICARUSF, non-linearities are allowed to be in the problem in the form of temperature dependent boundary parameters or temperature dependent material properties. In the computer program ICARUSB, the only non-linearity allowed is from the radiative boundary condition. Iterating within each time cycle is allowed to accommodate strong non-linearities.

boundary - In a problem their are two boundaries, one located at the minimum nodal point and one located at the maximum nodal point. These boundaries signify actual physical limits of the solid material in the problem, and are places where boundary conditions are applied to account for external influences on the body.

left or lower - Associated with the decreasing index and coordinate direction.

right or upper - Associated with the increasing index and coordinate direction.

4.0 THEORY

4.1 Energy Transport Equation

The heat transfer formulation is based on a control volume approach. Considering a control volume of material, the first law of thermodynamics for conservation of energy on that control volume can be written

$$\frac{\partial}{\partial t} \int_v \rho e dv + \int_s (q \cdot n)\, ds = \int_v u''' \, dv \qquad (4.1)$$

where e is the specific internal energy, t is time, q is the heat flux at the surface of the control volume (inward being positive), n is the outward facing normal at the control volume surface, u''' is the volumetric internal heat generation in the control volume, ρ is the density, and v and s denote volume and surface integrations respectively.

Using average quantities within the volume integrals, equation 4.1 becomes for a one-dimensional control volume

$$\rho c V \frac{\partial T}{\partial t} + [(qA)_r - (qA)_l] = u''' V \qquad (4.2)$$

where it has been assumed that de = cdT, where T denotes temperature, A is the surface area, V is the volume of the control volume, c is the specific heat, and r and l denote the right and left surfaces of the control volume respectively. The heat fluxes are obtained from Fourier's Law of Conduction,

$$q = - k \nabla T \qquad (4.3)$$

where k is the thermal conductivity and ∇T denotes the spatial gradient of temperature.

4.2 Gap Model

Material gaps may exist inside a problem where radiation and convective heat transfer are the energy transport mechanisms across the gap. The energy fluxes applied at the left and right edges of the gap are

$$Q_l = \sigma A_l (F_g)_{l-r}(T_l^{\,4} - T_r^{\,4}) + h_l A_l (T_l - T_b)$$

and (4.4)

$$Q_r = \sigma A_r (F_g)_{r-l}(T_l^{\,4} - T_r^{\,4}) + h_r A_r (T_b - T_r)$$

where σ is the Stefan-Boltzmann constant, F_g denotes the gray body form factor, h is the convection film coefficient, A is the area of the gap surfaces, T_b is the bulk temperature of the fluid in the gap, and the subscripts l and r denote the left and right edges of the gap respectively. The subscripts l-r and r-l denote the gray body factors for the left side viewed from the right and the right side viewed from the left, respectively.

The gray body form factor where the geometry is either parallel flat plates, concentric cylinders or concentric spheres and where the spatial dimension increases from left to right (thus $F_{l-r}=1$) is given by

$$(F_g)_{l-r} = \frac{1}{\dfrac{1}{\varepsilon_l} + \dfrac{A_l}{A_r}\left(\dfrac{1}{\varepsilon_r} - 1\right)} \qquad (4.5)$$

where ϵ is the emissivity. From reciprocity we also have

$$(F_g)_{r-1} = (F_g)_{1-r} \frac{A_1}{A_r} \tag{4.6}$$

The bulk convection temperature (T_b) is found by stipulating that the heat capacity of the convecting fluid is negligible. Thus,

$$h_1 A_1 (T_1 - T_b) = h_r A_r (T_b - T_r) \tag{4.7}$$

At the gap edge cells, a half cell energy balance is performed with the prescribed flux given by equation 4.4 used to replace one of the heat fluxes in the energy balance equation (equation 4.2).

4.3 Boundary Treatment and Boundary Conditions

The general boundary conditions available are:

1. A prescribed temperature at the boundary
2. A prescribed heat flux at the boundary
3. A general mixed condition that accounts for a combination of convection, radiation and applied flux at the boundary.

For an applied temperature condition (type 1) the boundary temperature is fixed in the solution scheme. For applied energy conditions (types 2 and 3) a half cell energy balance is performed. At the boundary surface, the prescribed energy flux is applied to replace one of the fluxes in the energy balance equation (4.2). Details of the flux representation are given in Section 5.1 and 6.1.

4.4 Numerical Solution Method

The numerical algorithm incorporates both implicit and explicit formulations. Equation 4.2 can be written in the form

$$\rho c V \frac{\partial T}{\partial t} + \theta[(qA)_r - (qA)_l]^{n+1}$$

$$+(1-\theta)[(qA)_r - (qA)_l]^n = u'''V \tag{4.8}$$

where $q^n = -(k\nabla T)^n$

and $q^{n+1} = -(k\nabla T)^{n+1}$

In equation 4.8, n and n+1 denote the old and new time levels in the time integration, and θ is a solution flag. Appropriate values of θ are 1.0 for an implicit solution and 0.0 for an explicit solution.

Simple first-order finite difference approximations are used for the gradients in the heat flux expressions and the time derivative. The spatial gradients are represented as

$$(\nabla T)_l = \frac{T_i - T_{i-1}}{z_i - z_{i-1}}$$

and (4.9)

$$(\nabla T)_r = \frac{T_{i+1} - T_i}{z_{i+1} - z_i}$$

The time derivate is approximated using backward differencing as

$$\frac{\partial T}{\partial t} = \frac{T_i^{n+1} - T_i^{n}}{\Delta t} \tag{4.10}$$

In equations 4.9 and 4.10, T denotes temperature, z denotes the spatial coordinate, ∇t is the timestep size. The subscripts i, i+1, and i-1 are the difference grid indices with node i representing the node about which the heat balance is occurring. The superscripts n and n+1 denote old and new time level values. Using these approximations, equation 4.8 can be cast in the form

$$B_i T_i^{n+1} = A_i T_{i+1}^{n+1} + C_i T_{i-1}^{n+1} + D_i \tag{4.11}$$

where A_i, B_i, and C_i are the coefficients that appear in front of the respective new time level temperature variables in the equation for node i, and D_i contains all contributions from source terms and old time level terms. Equation 4.9 is in the classic tri-diagonal form which lends itself to simple, efficient reduction. The equation solver described by Roach [2] is used in the reduction.

5.0 ICARUSB - AN INTERPRETIVE BASIC HEAT CONDUCTION PROGRAM

5.1 Program Capabilities

ICARUSB is a reduced version of the FORTRAN program ICARUS [1] that has existed on the main-frame computers at LLNL for a number of years. The program is maintained in intrepretive BASIC to provide a package that is usable on any IBM-PC like micro-computer as well as modifiable by the user.

The general capabilities of the computer program are:

1. Multiple material regions. There may be up to ten (10) different geometric or material regions in the problem.
2. Transient or steady-state solutions, with a choice of an implicit or explicit solution approach for transient problems.
3. Material gaps across which radiative and convective heat transfer can take place.

Limitations of the computer program are:

1. Constant material properties. The material properties may not vary with time or temperature.
2. Constant boundary condition parameters. The specifiable parameters at the boundary may not vary with time or temperature.

There are two types of boundary conditions available in the computer program ICARUSB:

1. A specified surface temperature that is held constant at all times.
2. A general condition accounting for radiation, convection and an applied heat flux. The total flux per unit area at the surface is represented by the equation

$$q_s = q_r - h(T_s - T_O) - \sigma(F_g)_{s-\infty} (T_s^4 - T_\infty^4) \qquad (5.1)$$

where h is the film coefficient, T_O is the reference bulk convection temperature,T_s is the surface temperature, q_r is the applied surface flux, F_g is the gray body form factor supplied by the user (see eq. 6.2), T_∞ is the reference body temperature, and σ is the Stefan-Boltzmann constant. The user supplies values for each of these parameters. For input purposes, a flux into the body is positive.

5.2 Program Structure

The intrepretive BASIC program is organized into a main block and 11 overlays. The main block and overlays are each an individual file on the disk. Each overlay consists of several blocks of coding that are accessed as subroutines. An overlay flow chart is given in Figure 5.1. The overlay and block functions are described below, and major variables in the program defined in Appendix B.

File	Block	Description
MAIN.BAS		performs variable dimensioning, controls the overall sequence
OLAY11.BAS	calcon	controls the actual calculation sequence
OLAY12.BAS	setcon	controls the setup sequence
OLAY21.BAS	blank	routine for inserting blank lines on the screen
	vdecode	decodes a string answer containing more than one value and breaks it into appropriate blocks

	chkchar	determines if the proper character response to a question has been made
	chkval	determines if an input value is within specified bounds
OLAY22.BAS	sblnk	inserts blank lines on the screen
	gblnk	writes blank lines to a specified output device
OLAY31.BAS	setio	controls writing and reading of the input record file
OLAY32.BAS	setchg	controls alteration of the input values
	bndchg	controls alteration of boundary parameters
OLAY33.BAS	setinit	controls initial input when the user choses to specify input from the terminal
	setmat	controls material property specification
	setgap	controls specification of gap parameters
OLAY34.BAS	initial	initializes some calculation variables before the actual calculation begins
	areavol	evaluates the volumes of the right and left halves of the cell and the areas at the right and left edges of the cell
	setedit	prints the setup record to the selected output device
	bndedit	controls editing of boundary parameters
	seditcon	controls disposition of the setup edit

	integ	controls the time integration
	bound	evaluates coefficients at the boundaries
	edit	performs a complete cycle edit
	editcon	controls disposition of cycle edit
	errchk	calculates the maximum absolute change between iterations
	coeff	evaluates the tri-diagonal coefficients at each interior node
	readtty	checks for a message from the terminal and responds accordingly
	trig	a general tri-diagonal solver
	writes	writes temperatures and heat fluxes to a disk file for use in the snapshot dump plots
	writet	writes temperatures and heat fluxes to a disk file for use in the time history plots
	flux	calculates heat fluxes
	energy	calculates internal energy
OLAY41.BAS	geomin	controls geometry region specification
	nodgap	determines the model type for each node, and the gap counter
	grid	loads the grid arrays
	boundin	controls specification of boundary condition parameters
	tempin	controls specification of temperature region parameters
	tstep	calculates the minimum explicit timestep
	gridcal	calculates mesh characteristics
OLAY42.BAS	tempfld	loads the temperature array during initialization
OLAY51.BAS		a dummy end block

5.3 Program Units

The computer program ICARUSB will operate with any set of consistent units. For problems involving radiation, an absolute temperature scale must be used, and the proper value for the Stefan-Boltzmann constant specified in the input. No units are given in the output, so the user must note and keep track of the units used in the calculation.

5.4 Program Usage

Because of the number of input and output files that can be connected to the program during execution, the user must enter BASIC by typing

```
BASICA /F:5
```

followed by a carriage return. The program is executed by placing the master disk in the default drive and typing

```
RUN"X:ICARUSB.BAS" (lf)
```

where X denotes the default drive letter and (lf) denotes line-feed. During execution, the user will be asked to make several decisions that affect the calculation sequence. The options are explained by the program, so no further discussion is needed at this time.

Timing information indicates that the execution speed is approximately 0.24 seconds/node-cycle. This is a factor of 12 greater than the value for the compiled FORTRAN program ICARUSF discussed in Section 6.

5.5 Output File Organization

Depending on user responses to control and input questions, as many as three files may be created during execution. The files and their default names are:

output.dat	This file contains the input record and requested edits generated during execution
isnap.dat	This file contains the temperature information necessary for generating snapshot plots during the post-processing phase
itime.dat	This file contains the temperature information necessary for generating time history plots during the post-processing phase

The files will be located on the drive specified by the user during execution. Depending on the problem, the files may be large so the user should ensure that there is ample disk space available.

5.6 Miscellaneous

In ICARUSB the total internal energy is calculated by summing for all cells the product of temperature and specific heat. The change in this quantity is the energy gains or losses at the system boundaries plus gains due to internal heat generation. The total internal energy is given in the full cycle edits as well as screen status edits.

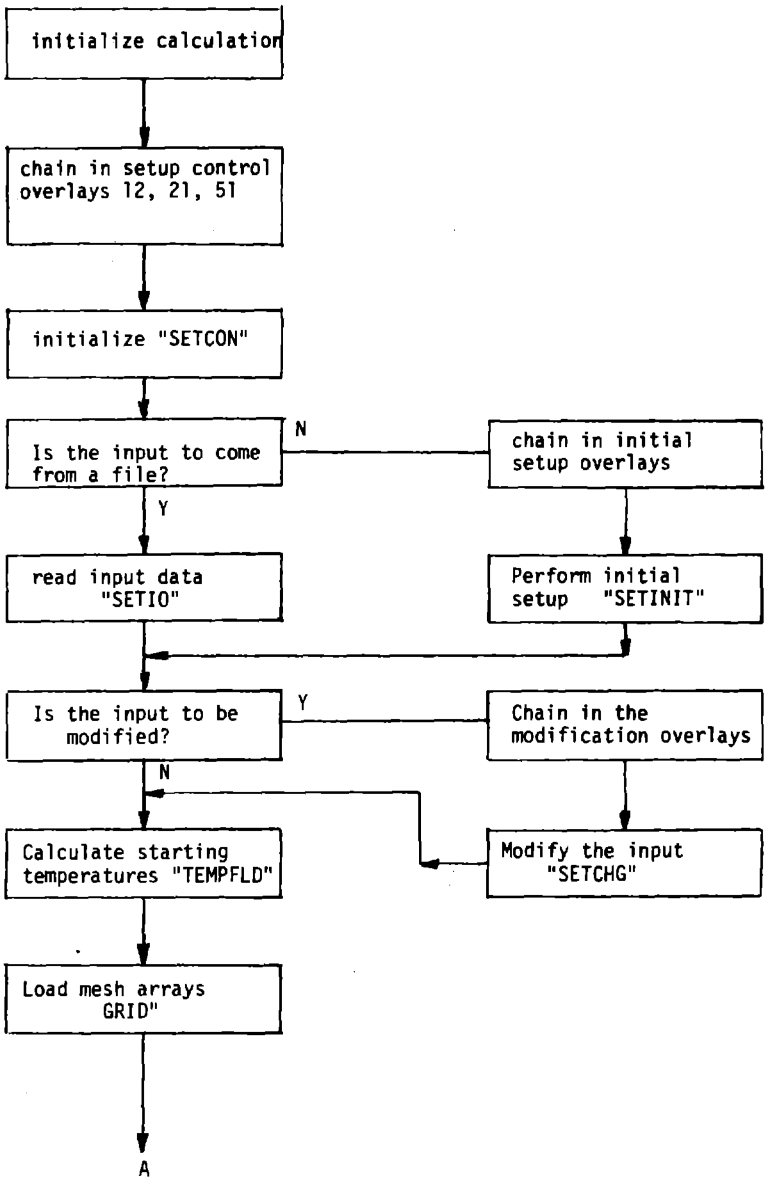

initialize calculation
chain in setup control overlays 12, 21, 51
initialize "SETCON"
Is the input to come from a file?
N
Y
chain in initial setup overlays
read input data "SETIO"
Perform initial setup "SETINIT"
Is the input to be modified?
Y
N
Chain in the modification overlays
Calculate starting temperatures "TEMPFLD"
Modify the input "SETCHG"
Load mesh arrays GRID"
A

Figure 5.1 - Flow chart of the interpretive BASIC heat conduction code ICARUSB.

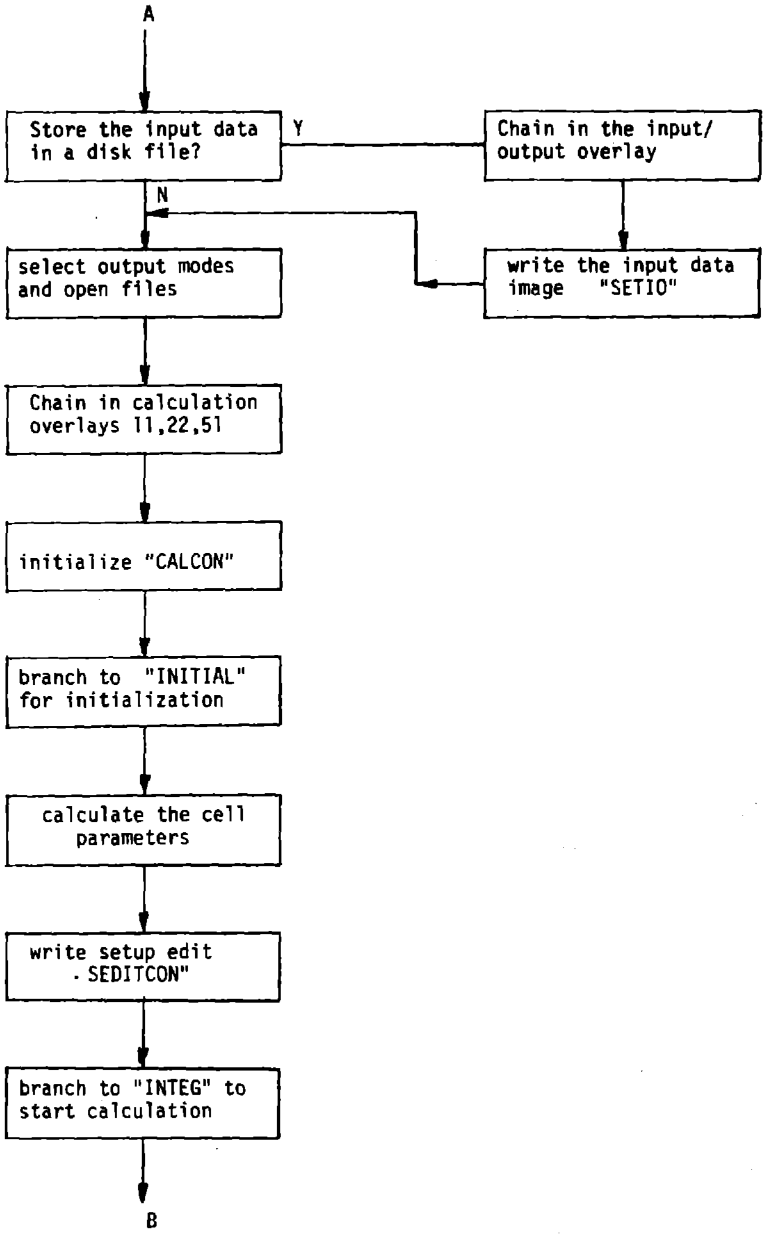

A
Store the input data in a disk file?
Y
N
Chain in the input/ output overlay
write the input data image "SETIO"
select output modes and open files
Chain in calculation overlays 11,22,51
initialize "CALCON"
branch to "INITIAL" for initialization
calculate the cell parameters
write setup edit . SEDITCON"
branch to "INTEG" to start calculation
B

Figure 5.1 - Continued

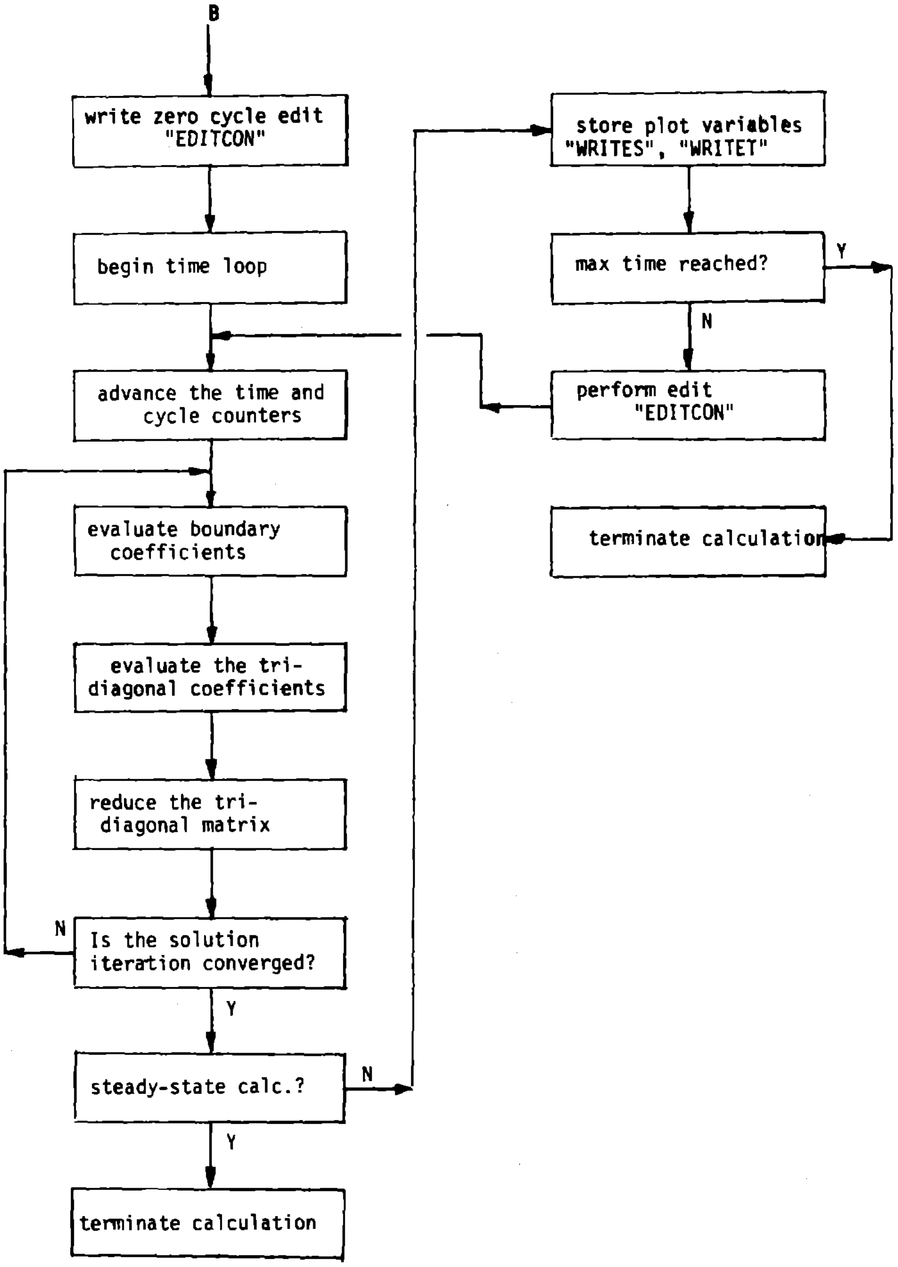

B
write zero cycle edit "EDITCON"
begin time loop
advance the time and cycle counters
evaluate boundary coefficients
evaluate the tri-diagonal coefficients
reduce the tri-diagonal matrix
Is the solution iteration converged?
N
Y
steady-state calc.?
N
Y
terminate calculation
store plot variables "WRITES", "WRITET"
max time reached?
Y
N
perform edit "EDITCON"
terminate calculation

Figure 5.1 - Continued

6.0 ICARUSF - A COMPILED FORTRAN HEAT CONDUCTION PROGRAM

6.1 Program capabilities

ICARUSF is a reduced version of the FORTRAN program ICARUS [1] that has existed on the main-frame computers at LLNL for a number of years. The program is maintained in a compiled FORTRAN form to provide a package that is as fast as can currently be obtained on the IBM-PC. The executable file was generated using version 3.2 of the Microsoft Fortran compiler [3].

The general capabilites of the compiled FORTRAN version of the computer program are:

1. Multiple geometry/material regions. There may be up to ten (10) different geometric or material regions in the problem.
2. Transient or steady-state solutions, with an implicit or explicit solution approach for transient problems.
3. Material gaps across which radiative and convective heat transfer can take place.
4. User specification of temperature dependent thermal conductivity and specific heat and time or temperature dependent internal heat generation.
5. User specification of time or temperature dependent boundary condition parameters.

There are three boundary types available in ICARUSF. The conditions are:

1. A specified surface temperature that may be a function of time.
2. A specified surface heat flux that may be a function of surface temperature or time.

3. A general condition accounting for radiation between the body surface and another (reference) surface, convection with a fluid, and applied heat flux at the surface. The total surface flux is represented by

$$q_s = q_r - h(T_s - T_o) - \sigma(F_g)_{s-\infty}(T_s^{\,4} - T_\infty^{\,4}) \qquad (6.1)$$

where

$$(F_g)_{s-\infty} = \frac{1}{(\frac{1}{\varepsilon_s} - 1) + \frac{A_s}{A_\infty}(\frac{1}{\varepsilon_\infty} - 1) + \frac{1}{F_{s-\infty}}} \qquad (6.2)$$

and q_r is the surface flux, h is the surface film coefficient, T_O is the reference bulk fluid temperature, is the Stefan-Boltzmann constant, A_s and A_∞ are the areas of the surface and reference surface, ϵ_s and ϵ_∞ are the emissivities of the surface and reference surface, T_s is the body surface temperature, T_∞ is the reference radiation temperature and $F_{s-\infty}$ is the radiation view factor. The quantities , q_r, h, T_O, and T_∞ can be specified as constants or as functions of time or the surface temperature by the user. The emissivities ϵ_s and ϵ_∞ are specifiable only as constants or functions of temperature. For input purposes, an applied flux into the body is positive.

In the specification of time or temperature dependent properties or boundary conditions, piecewise linear curves are used. As many as ten curves may be specified by the user. An example of such a curve, and the associated data is given in Figure 6.1. Each curve is limited to 20 entries.

6.2 Program Structure

The compiled FORTRAN program is organized into a main routine and 23 subroutines. The source is divided into 16 individual files. A calculation flow chart is given in Figure 6.2. The program structure and subroutine functions are described below and global variables defined in Appendix C.

File	Routine	Description
MAIN.FOR	main	- performs calculation initialization and file creation
	files	- checks for the existance of requested input and output files and takes appropriate action
ACONTRL.FOR	acontrl	- controls the calculation sequence
AVOL.FOR	areavm	- calculates the volume of the left half of the cell and the area at the left edge of the cell
	areavp	- calculates the volume of the right half of the cell and the area at the right edge of the cell
BLOCK.FOR	block	- contains default data statements
BOUND.FOR	bound	- evaluates boundary parameters and sets up the coefficients at the boundaries
	bound1	- determines the boundary parameters at the current time based on the time or surface temperature
COEFF.FOR	coeff	- calculates the tri-diagonal coefficients at each interior node
EDIT.FOR	edit	- controls all long edits written to the output file
	edit1	- controls all short edits written to the output file
	energy	- calculates the total internal energy and the minimum and maximum temperatures
ERROR.FOR	error	- calculates the non-linear iteration convergence error at each iteration
PDUMP.FOR	sdump	- writes information for the snapshot plot dump
	tdump	- writes information for the time history plot dump

	flux	- calculates the heat flux at a specified nodal point
PROP.FOR	prop	- controls evaluation of all property values
	prop1	- evaluates conductivity and specific heat as functions of temperature, and internal heat generation as a function óf time or temperature
	eprop1	- performs the table look-up for property evaluation
RTTY.FOR	readtty-	reads control messages from the user terminal to control status responses and invoke early termination if requested
SETUP.FOR	setup	- reads all input data and initializes all arrays before execution is commenced
SOFF.FOR	signoff-	writes the final termination messages
TRIG.FOR	trig	- a general tri-diagonal equation solver
TSTEP.FOR	tstep	- calculates the current explicit timestep
ZERO.FOR	zero	- zero’s out memory

The subroutine "readtty" executes the routine "inkey" contained in the library P-PLOT [4]. The routine "inkey" accesses the terminal message buffer. The library P-PLOT is used at the Lawrence Livermore National Laboratory under a site licensing agreement. To load the program to produce the executable ICARUSF.EXE, it is necessary to name elements of the P-PLOT library on the load command line. Further information on this subject can be obtained from the author.

6.3 PREICAR - An Interpretive BASIC Pre-Processor

To facilitate input from the user terminal, a compiled BASIC pre-processor was written. The executable file was created using the IBM BASIC compiler [5]. The program is organized into a main program and 21 subroutines. A flow chart of the computer program is given in Figure 6.3. The functions of the various blocks are given below and major variables are defined in Appendix D.

Block	Description
main	contains all dimension statements and controls the overall setup sequence
setinit	controls specification of initial input by the user
setchg	controls alteration of the user input
bmod	controls modification of the boundary parameter values
blank	inserts blank lines on the screen
vdecode	decodes a string answer containing more than one value and breaks it into appropriate blocks
incurv	controls specification of the piecewise linear curves
geomin	controls geometry region specification
ngap	determines the number of gaps in the problem
setgap	controls specification of gap parameters
setbnd	controls initial specification of boundary parameters
setmat	controls the initial specification of material properties
setio	controls writing and reading of the input record file
tempin	controls specification of temperature region parameters
addcurv	controls addition of a piecewise linear curve during input file modification
curvmod	controls modification of piecewise linear curve entries
check	checks letter response for correctness
chklim	determines if a response is within specified limits
cntcurv	sets curve specification flags and determines the number of curves referenced

gridcal calculates the left and right mesh spacing for each geometric region

error traps errors and prints out error number and line where error occured

ckfile determines if a file exists and is of the proper format

6.4 Program Units

The computer programs (ICARUSF and PREICAR) will operate with any set of consistent units. For problems involving radiation, an absolute temperature scale must be used, and the proper value for the Stefan-Boltzmann constant specified in the input. No units are given in the output, so the user must note and keep track of the units used in the problem.

6.5 Program Usage

PREICAR:

The pre-processor program (file PREICAR.EXE) can be executed by entering BASIC, loading the master disk in one of the computer drives, and typing

```
X:PREICAR (lf)
```

where X denotes the drive containing the master disk and (lf) denotes line-feed. During execution the user will be prompted for file locations and names. The executable file (PREICAR.EXE) was generated using version 1.0 of the IBM BASIC Compiler [5] which does not support directory structure. Thus, all files must reside in the root directory level.

ICARUSF:

The main calculation program can be executed by loading the master disk in one of the computer drives and typing

X:ICARUSF (lf)

where X denotes the drive containing the master disk, and (lf) denotes line-feed. The user will be prompted for some input and output file name information before actual execution begins. Microsoft FORTRAN [3] does not support directory structure, so all files to be accessed must be in the default directory for the drive. During execution a status message giving current cycle and time as well as other parameters may be obtained. The method for obtaining this edit is explained by the code during execution.

Timing information indicates that the execution speed is approximately 0.02 seconds/node-cycle using the 8087 co-processor. This is a factor of 12 faster than the interpretive BASIC program ICARUSB discussed in Section 5.

6.6 Output File Organization

PREICAR:

The only file produced by the pre-processing program is the file to be used as input to the main heat conduction program ICARUSF. During execution of PREICAR, the user is requested for a location and name for this file. As discussed previously, directory structure is not allowed in the name specification.

ICARUSF:

Depending on user responses to input questions and control questions, as many as three output files will be generated during execution. The default names and functions of the files are:

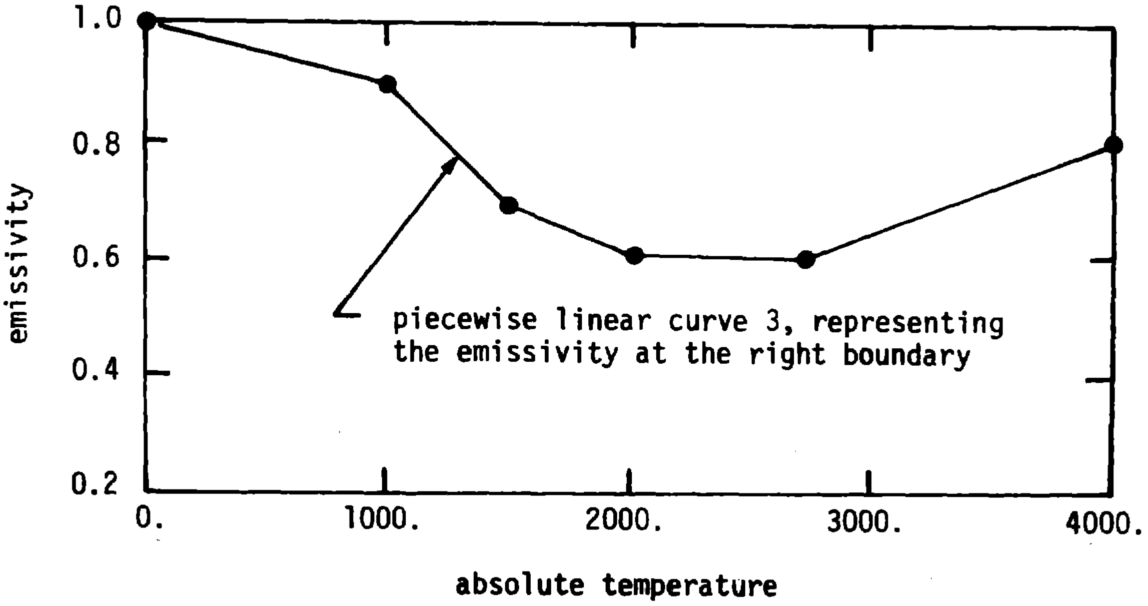
output.dat	This file contains the input record and full edits at the requested times.

isnap.dat This file contains the temperature information necessary to produce the snapshot plots during post-processing.

itime.dat This file contains the temperature information necessary to produce the time history plots during post-processing.

6.7 Miscellaneous

In ICARUSF the internal energy of the system is calculated by summing for all cells the product of temperature and specific heat. For a constant property material this is a precise representation of the total internal energy and represents the energy gains and losses at the boundaries plus gains due to internal heat generation. For the case of temperature dependent specific heat, this is only an approximate representation of the boundary heat transfer and internal heat generation because the true internal energy is the integral over temperature of the specific heat. The total energy is given in the full cycle edits as well as screen status edits.



Temperature	Emissivity
0.	1.0
1000.	0.9
1500.	0.7
2000.	0.6
2750.	0.6
4000.	0.8

Figure 6.1 - An example of a piecewise linear curve.

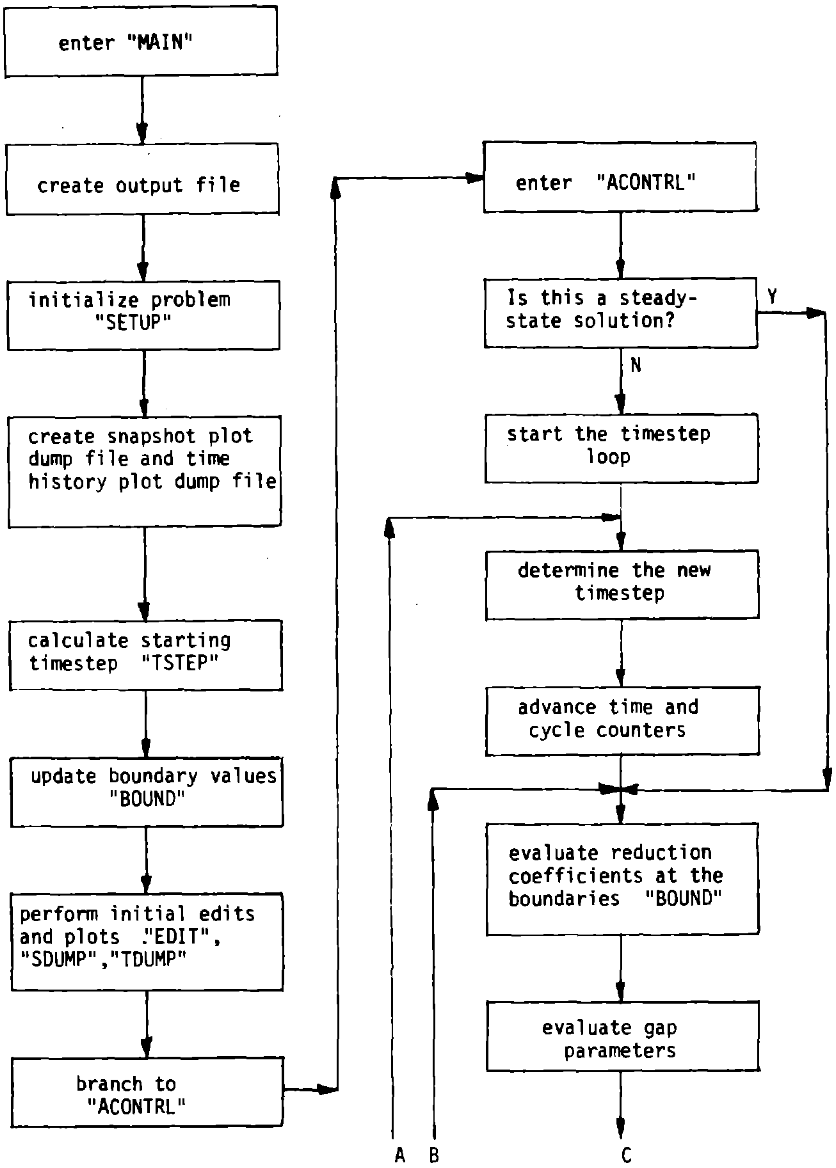

enter "MAIN"
create output file
initialize problem "SETUP"
create snapshot plot dump file and time history plot dump file
calculate starting timestep "TSTEP"
update boundary values "BOUND"
perform initial edits and plots ."EDIT", "SDUMP","TDUMP"
branch to "ACONTRL"
enter "ACONTRL"
Is this a steady-state solution?
Y
N
start the timestep loop
determine the new timestep
advance time and cycle counters
evaluate reduction coefficients at the boundaries "BOUND"
evaluate gap parameters
A
B
C

Figure 6.2 - Flow chart of the compiled FORTRAN heat conduction code ICARUSF.

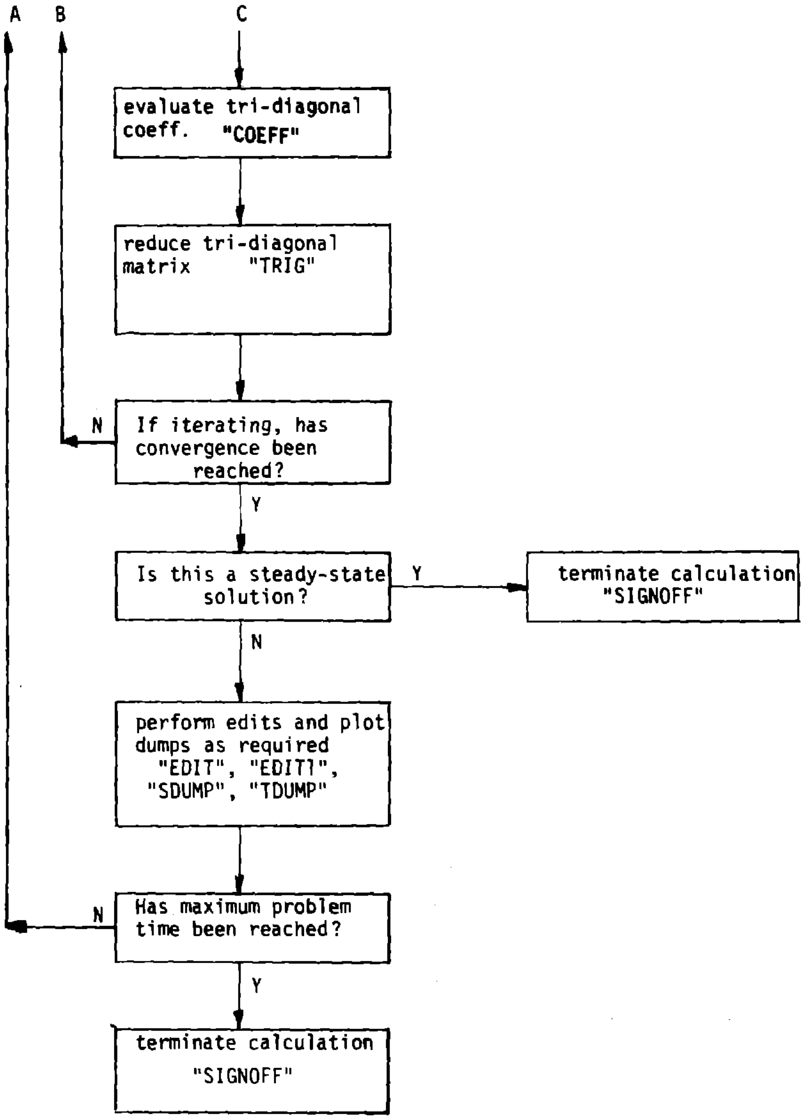

A
B
C
evaluate tri-diagonal coeff. "COEFF"
reduce tri-diagonal matrix "TRIG"
If iterating, has convergence been reached?
N
Y
Is this a steady-state solution?
Y
terminate calculation "SIGNOFF"
N
perform edits and plot dumps as required "EDIT", "EDIT1", "SDUMP", "TDUMP"
Has maximum problem time been reached?
N
Y
terminate calculation "SIGNOFF"

Figure 6.2 - Continued

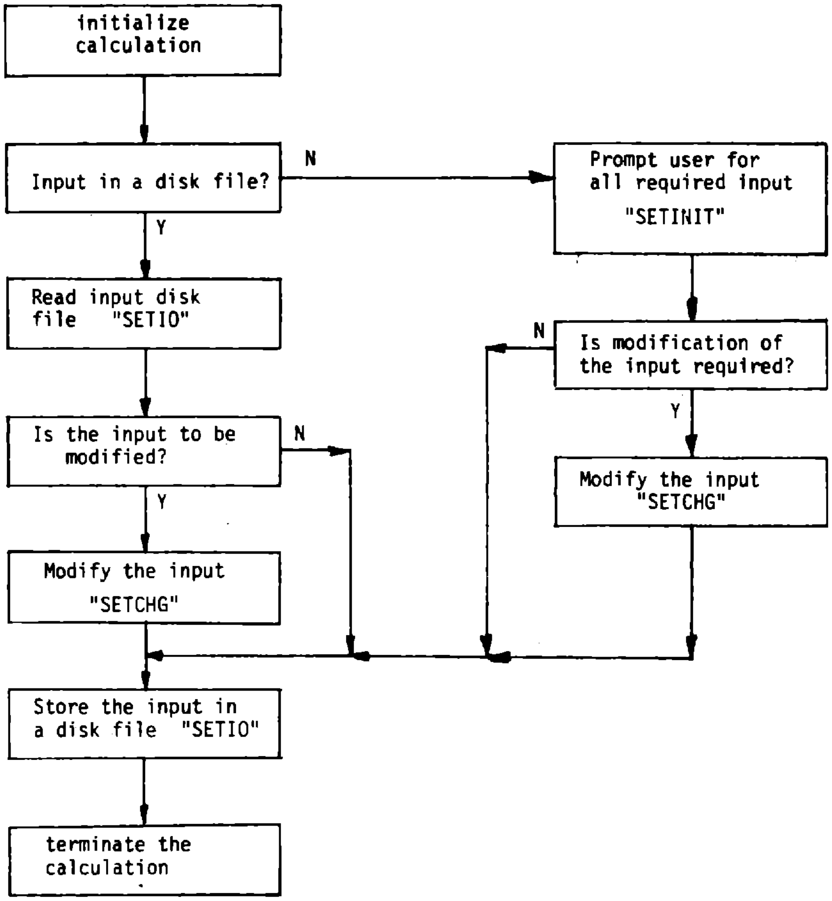

initialize calculation
Input in a disk file?
N
Y
Prompt user for all required input "SETINIT"
Read input disk file "SETIO"
N
Is modification of the input required?
Y
Is the input to be modified?
N
Y
Modify the input "SETCHG"
Modify the input "SETCHG"
Store the input in a disk file "SETIO"
terminate the calculation

Figure 6.3 - Flow chart of the interpretive BASIC pre-processing code PREICAR.

7.0 PLOTIC AND CPLOTIC - BASIC PLOT POST-PROCESSING PROGRAMS FOR ICARUSB AND ICARUSF

7.1 Program Capabilities

PLOTIC and CPLOTIC are BASIC programs that produce plots from data generated during the execution of the heat conduction programs ICARUSB, ICARUSF and PREICAR. Three types of plots can be produced:

1. Spatial plots of temperature or heat flux. Data from the snapshot plot dump file produced by ICARUSB or ICARUSF is used for the plot.

2. Time history plots of temperature or heat flux. Data from the time history plot dump file produced by ICARUSB or ICARUSF is used for the plot.

3. Grid plots of the finite difference mesh. Information contained in the heat transfer code input file produced by ICARUSB or PREICAR is used for the plot. A small circle is placed at nodes tagged for time history storage.

The programs allow the user to select the type of plot, and the specific time for a spatial plot or node for a time history plot.

PLOTIC is an interpretive BASIC routine. CPLOTIC is a compiled BASIC routine that differs in two ways from the interpretive BASIC routine PLOTIC:

1. In interpretive BASIC the user is allowed to use directory structure in the specification of file names. This feature is not supported by Version 1.0 of the IBM BASIC compiler [5] which was used to generate CPLOTIC. Thus, directory structure can be used in PLOTIC but cannot be used in CPLOTIC.

2. Both programs use a small assembly language routine to perform a "shift-prtsc" (printing of screen contents) from inside the program. The form of the statement required to execute this routine is different for use with a compiler than for the interpretive mode.

Some clarification of the heat flux representation is required. At internal nodes, the heat flux is represented by Fourier's Law (equation 4.3). At all gap surfaces the plotted heat flux is given by equations 4.4. At problem boundaries, for specified temperature the flux is calculated using a one sided differencing of Fourier's Law (equation 4.3). For applied flux conditions, equation 5.1 or 6.1 is used to evaluate the flux. In the plots, a positive flux denotes left to right heat flow. Thus a flux into the body at the right boundary will be plotted as negative.

7.2 Program Structure

Both BASIC programs are organized into a main block and 13 subroutines. A flow chart is given in Figure 7.1 and definitions of major variables given in Appendix E. The functions of the various subroutines are:

Block	Description
main	contains dimension statements and controls the overall plotting process
spatial	controls generation of the spatial snapshot plots of temperature or heat flux
temporal	controls generation of the temporal plots of temperature or heat flux at specified nodal points
map	draws the coordinate grid and writes coordinate labels
scale	determines the scale factors, corrected plot limits and number of tick marks and numbers on each axis

grid	controls generation of the mesh plots
sdump	reads the spatial snapshot plot dump
tdump	reads the time history plot dump
idump	reads the heat transfer code input file
blank	inserts blank lines on the screen
vdecode	decodes a string answer containing more than one value and breaks it into appropriate blocks
action	controls plot disposition
chkfile	checks for existance of a file and determines if it is of the proper format
error	performs error trapping diagnostic

7.3 Input File Organization

The data bases for the two plot dump files generated by the heat conduction programs ICARUSB and ICARUSF are significantly different. However, since the form of the data base is unimportant to the user they will not be discussed.

7.4 Program Units

The computer programs PLOTIC and CPLOTIC use data generated by the programs ICARUSB, ICARUSF and PREICAR. The time, space and temperature units used in the programs are not altered. Units are not indicated on the plots, so the user must note and keep track of units.

7.5 program Usage

PLOTIC:

To execute the program, first enter BASIC, then place the master disk in a disk drive and type

```
RUN"X:PLOTIC.BAS" (lf)
```

where X denotes the drive containing the master disk and (lf) denotes line-feed. The user will be prompted for the names of the files containing the snapshot data or time history data or mesh data after a plot type has been requested.

CPLOTIC:

To execute the program [file CPLOTIC.EXE], place the master disk in a disk drive and type

```
X:CPLOTIC (lf)
```

where X denotes the drive containing the master disk and (lf) denotes line-feed. The user will be prompted for the names of the files containing the snapshot data or time history data after a plot type has been requested. Note that directory structure cannot be used in designating the location of files in CPLOTIC.

General:

During execution the plots are produced on the screen. The procedure to obtain a hard copy of the plot on a dot matrix printer is described on the screen under the plot. Please note that the appropriate graphics package must have been loaded into the computer prior to executing the program if hard copies of the plots are desired. For example, on the IBM PC the DOS file GRAPHICS.COM must be loaded into the system. To produce a hard copy of a plot on a dot matrix printer will take several minutes because of the translation required by the operating system's resident graphics package.

Examples of the plots that can be produced are given in Figures 7.2, 7.3 and 7.4

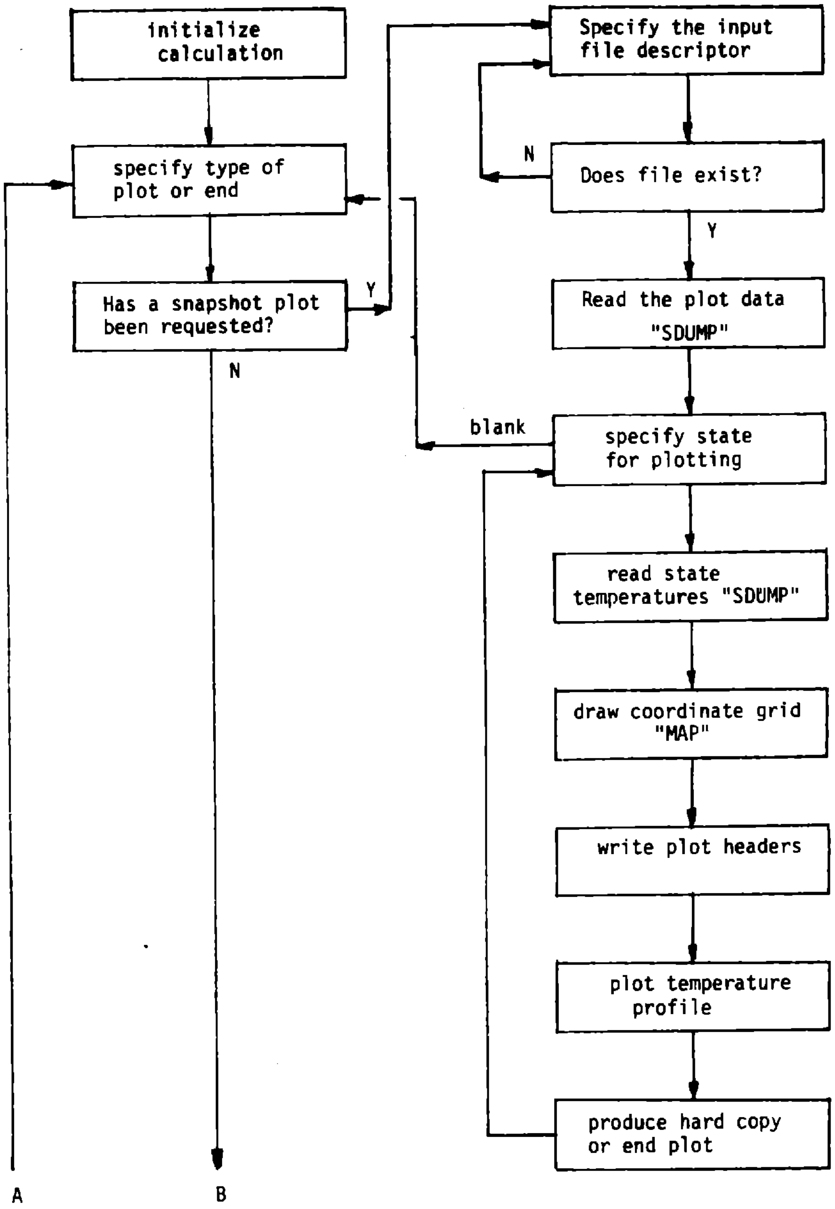
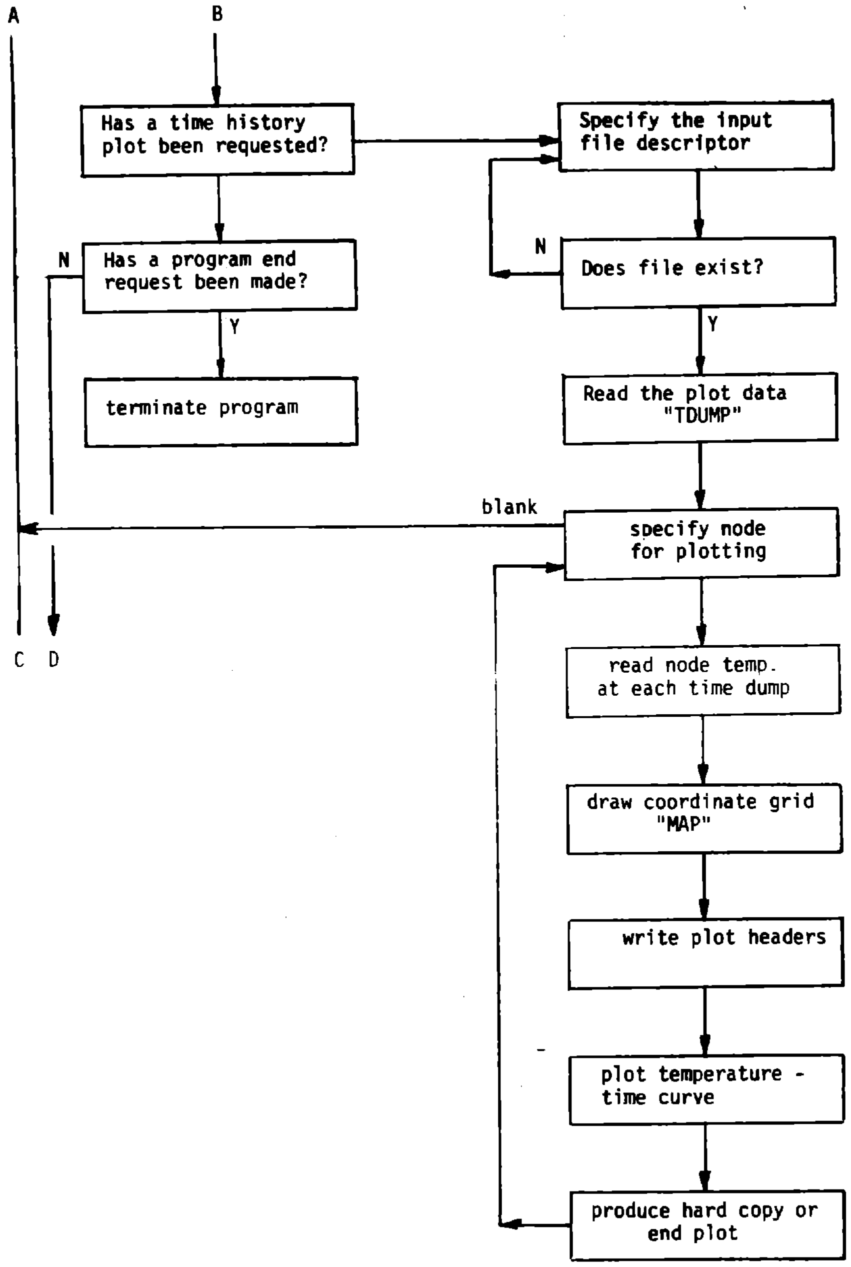

initialize calculation
specify type of plot or end
Has a snapshot plot been requested?
Y
N
Specify the input file descriptor
Does file exist?
N
Y
Read the plot data "SDUMP"
blank
specify state for plotting
read state temperatures "SDUMP"
draw coordinate grid "MAP"
write plot headers
plot temperature profile
produce hard copy or end plot
A
B

Figure 7.1 - Flow chart of the interpretive BASIC plot code PLOTIC and the compiled BASIC plot code CPLOTIC.



A
B
Has a time history plot been requested?
Specify the input file descriptor
N
Has a program end request been made?
Y
N
Does file exist?
Y
terminate program
Read the plot data "TDUMP"
blank
specify node for plotting
C
D
read node temp. at each time dump
draw coordinate grid "MAP"
write plot headers
plot temperature - time curve
produce hard copy or end plot

Figure 7.1 - Continued

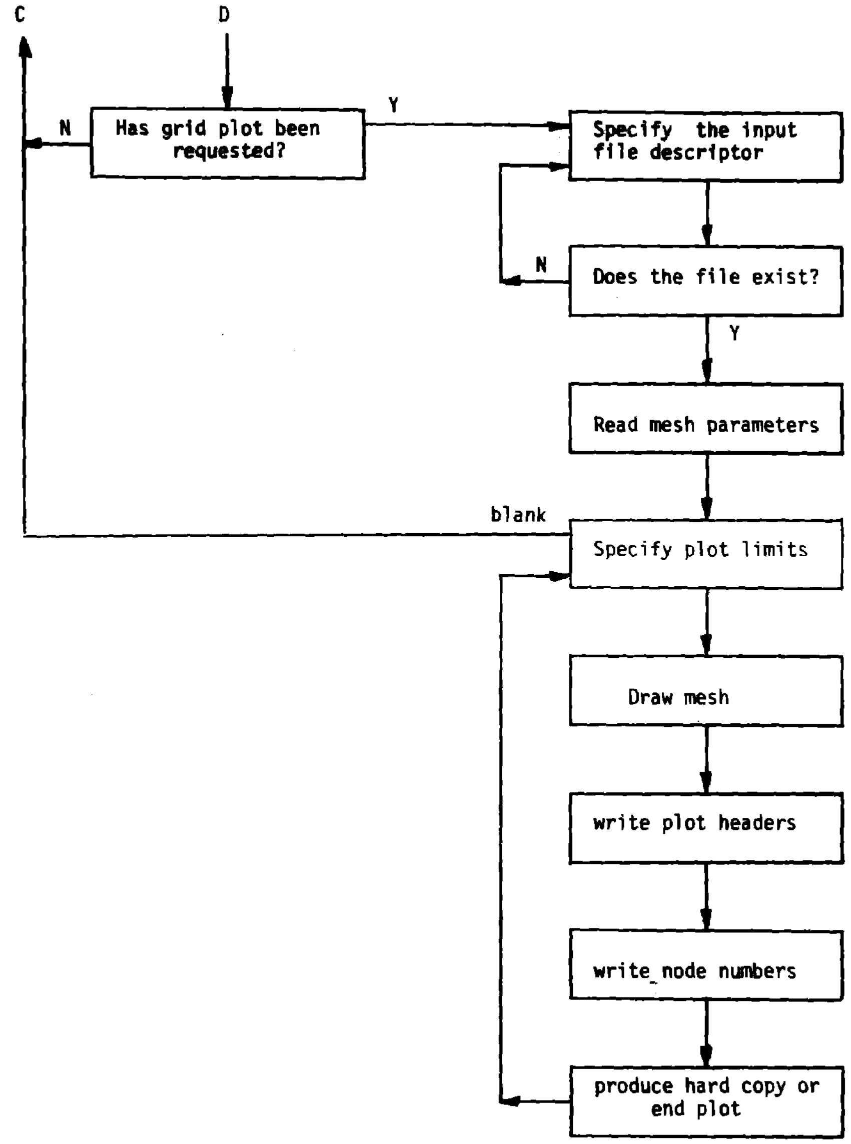


Figure 7.1 - Continued

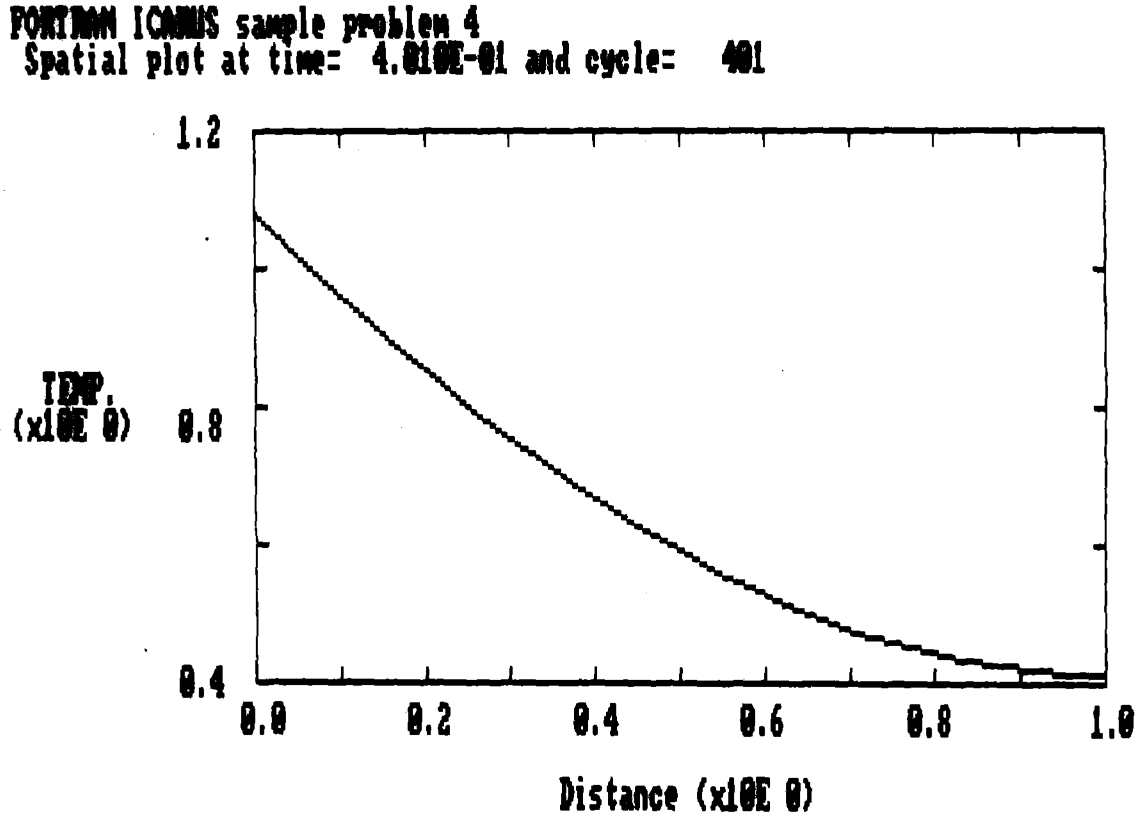


Figure 7.2 - An example of a spatial plot produced by the post-processing plot codes PLOTIC and CPLOTIC.

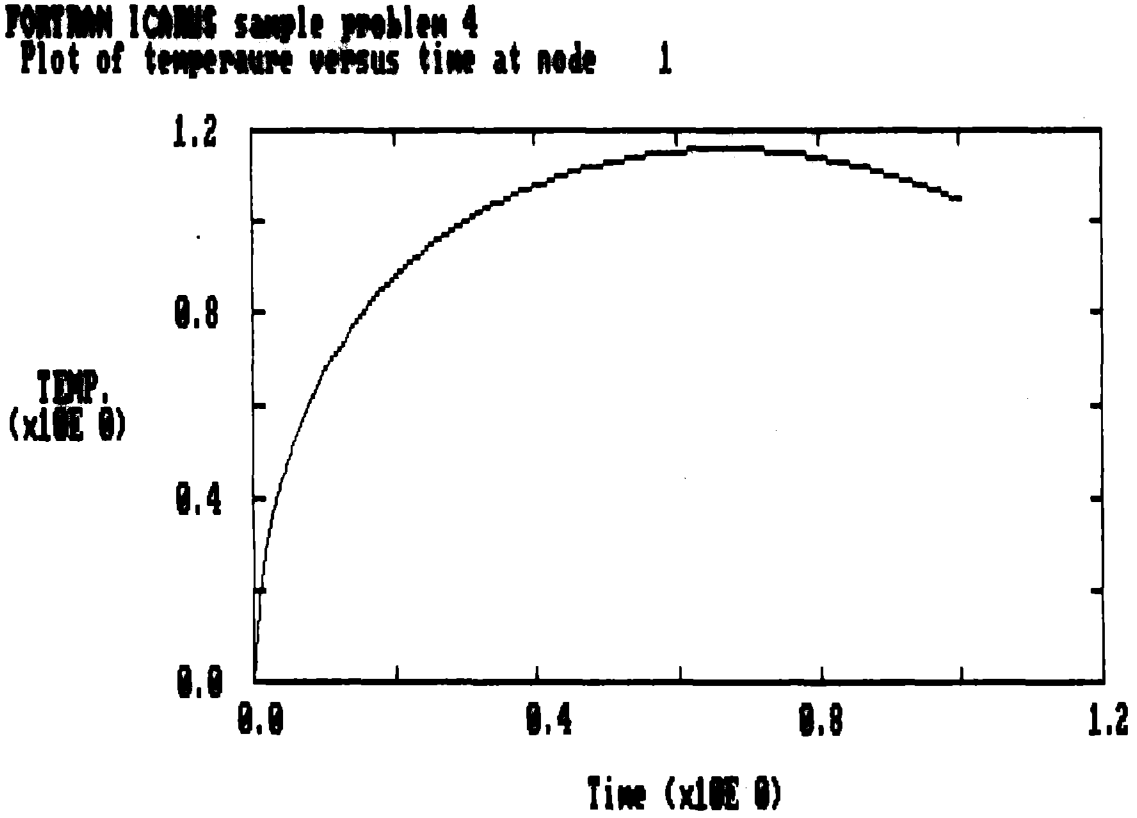


Figure 7.3 – An example of a time history plot produced by the post-processing plot codes PLOTIC and CPLOTIC.

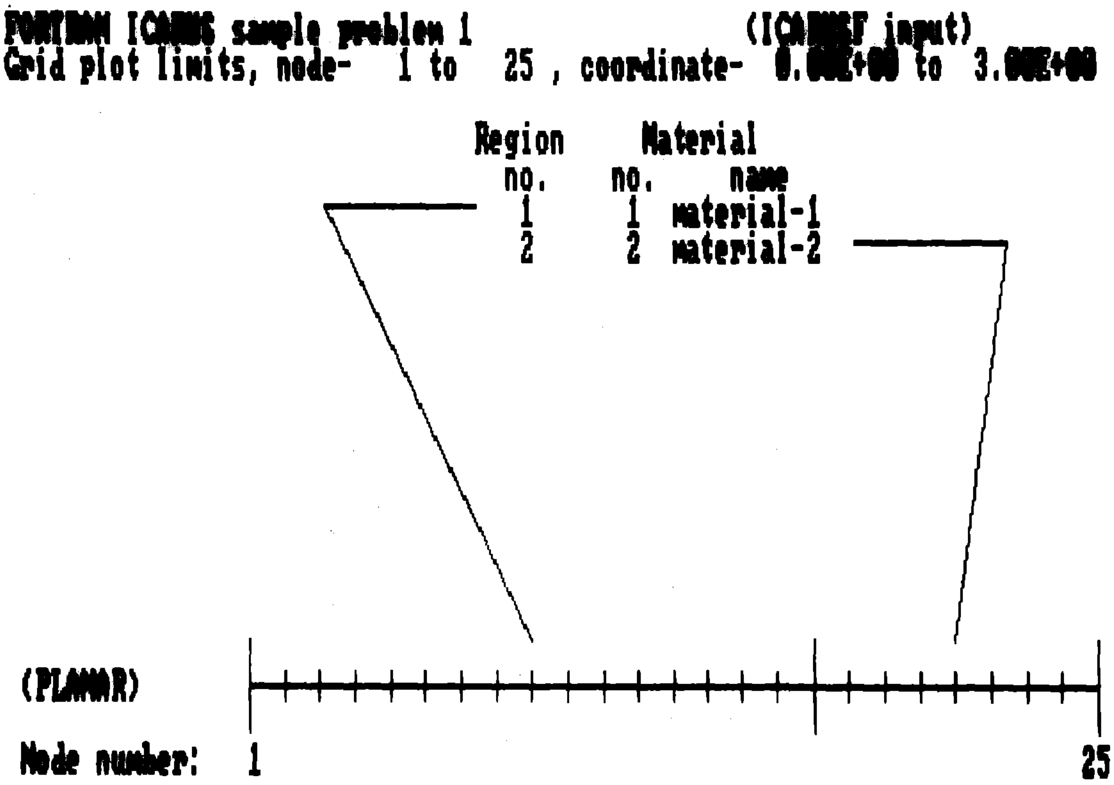

FORTRAN ICARUS sample problem 1
(ICARUSF input)
Grid plot limits, node- 1 to 25 , coordinate- 0.00E+00 to 3.00E+00
Region no.
Material no. name
1 1 material-1
2 2 material-2
(PLANAR)
Node number: 1
25

Figure 7.4 - An example of a grid plot produced by the post-processing plot programs PLOTIC and CPLOTIC.

8.0 SAMPLE PROBLEMS

To demonstrate the capabilities of the computer program, and verify the models and solution algorithm, a series of verification calculations were performed. Seven of the problems are reported in this section. The results predicted by the ICARUS programs are compared to other computer program predictions and/or analytical solutions.

8.1 Transient Conduction in a Two Material Region Solid with Convective Heating (Sample Problem 1)

This problem calculates heat conduction in a region containing two materials, and having convective heating on one boundary and an insulated condition on the other boundary. The geometry and material properties are given in Figure 8.1. There are 24 uniform grid spacings in the problem. In Table 8.1, the predicted temperature behavior is compared to the analytical solution taken from Schneider [6]. With the course (unoptimized) grid used, there is at most a 5 percent difference, at early times, between the program prediction and the analytical prediction.

8.2 Steady-State Conduction in a Body with Specified Boundary Temperatures (Sample Problem 2).

This problem determines the steady-state profile in a single material problem with specified temperatures on the boundaries. The geometry and material properties are given in Figure 8.2. Twenty uniform mesh spacings were used in the problem. The analytical solution to this problem is

$$T = -x + 1 \qquad (8.1)$$

where T is temperature and x is the spatial distance measured from the left surface. The program prediction matched exactly the analytical solution out to the five significant figures printed.

8.3 Transient Condution with a Radiative Boundary (Sample Problem 3)

This problem duplicates a single material planar system described by Schneider [6]. The geometry, material properties and boundary conditions are given in Figure 8.3a. There are 24 uniform mesh spacings in the problem. A comparison of the numerical predictions with the analytical solution is given in Figure 8.3b.

8.4 Transient Conduction with a Linearly Decreasing Surface Heat Flux (Sample Problem 4)

This problem determines the transient behavior of a planar body subjected on one boundary to a heat flux that decreases linearly with time, and to an insulated condition on the other boundary. The geometry, material properties and boundary conditions are given in Figure 8.4a. There are 24 uniform mesh spacings in the problem. Figure 8.4b gives a comparison of the numerical predictions with the analytical solution.

8.5 Transient Conduction in a Sphere with a Convectively Cooled Surface (Sample Problem 5)

This problem is an example of a solution in spherical geometry. The surface of the sphere is convectively cooled with a constant film coefficient. The geometry, material properties and boundary conditions are given in Figure 8.5a. There are 25 uniform mesh spacings in this problem. Figure 8.5b gives a comparison of the numerical prediction with the analytical solution graphically presented by Schneider [6].

8.6 Transient Conduction in an Infinite Solid Surrounding a Spherical Cavity (Sample Problem 6)

The geometry, material properties and boundary conditions are given in Figure 8.6a. There are 81 uniformly spaced nodal points in this problem. The surface of the spherical cavity is subjected to an applied temperature. Figure 8.6b gives a comparison of the numerical prediction with the analytical solution.

8.7 Transient Conduction in a Semi-infinite Solid with Non-linear Material Properties and Applied Surface Heat Flux (Sample Problem 7)

This problem was used by Shapiro [7] as a demonstration of the capabilities and accuracy of the two-dimensional finite element heat conduction program TOPAZ. The geometry, material properties and boundary conditions are given in Figure 8.7a. There are 51 nodal points in the problem. The nodal point spacing was graded using a factor of 1.11. The smallest mesh space, at the flux boundary, is 1.88×10^{-5}. A comparison of the ICARUS prediction with the TOPAZ prediction and the analytical solution is given in Table 8.2. Because the ICARUS algorithm assumes linear variation in temperature between the nodes, while the TOPAZ algorithm assumes linear variation in heat flux between the nodes (higher order approximation), a finer zoning was required in the ICARUS solution at the flux boundary where sharp temperature gradients result.

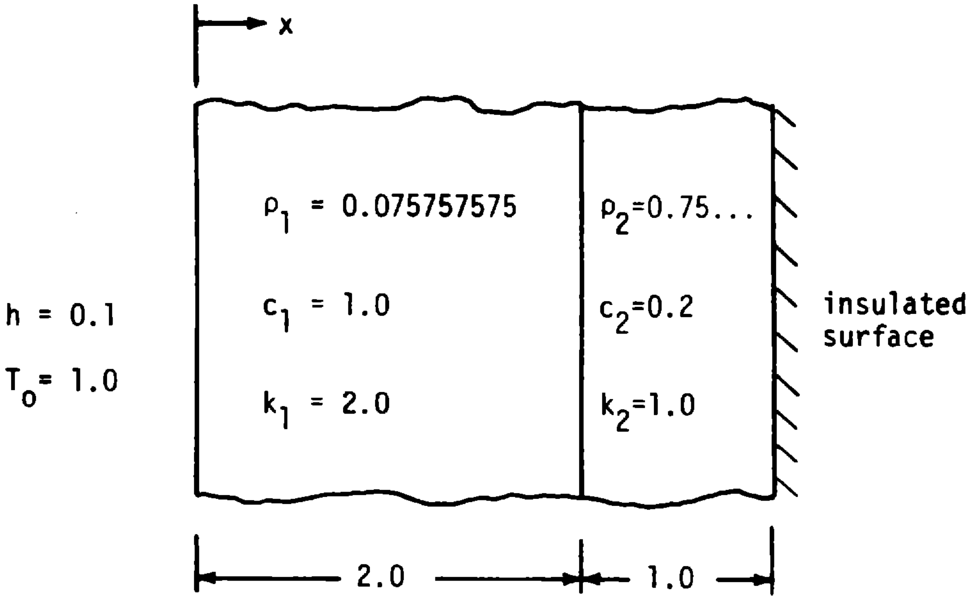


Figure 8.1 - Geometry and material properties for sample problem 1 - transient two-region heat conduction.

Table 8.1

Sample problem 1 - comparison of the code prediction to the analytical solution taken from chart 31 of Schneider [4].

Time	Node 1		Node 25	
	code	chart	code	chart
0.0	0.	-	0.	-
0.5	0.1979	-	0.1164	-
1.0	0.3126	-	0.2428	-
1.5	0.4117	0.39	0.3519	0.36
2.0	0.4958	0.48	0.4446	0.45
2.5	0.5680	0.55	0.5241	0.53
3.0	0.6297	0.62	0.5921	0.59
3.5	0.6827	0.68	0.6505	0.65
4.0	0.7281	0.72	0.7005	0.70

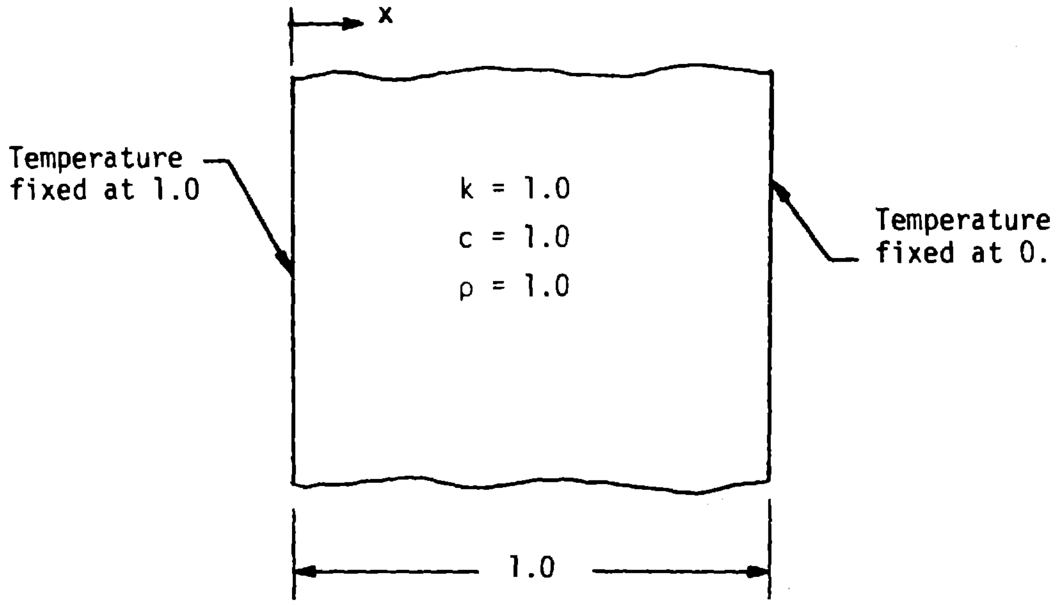


Figure 8.2 – Geometry and material properties for sample problem 2 – steady-state heat conduction in a region bounded with specified temperature.

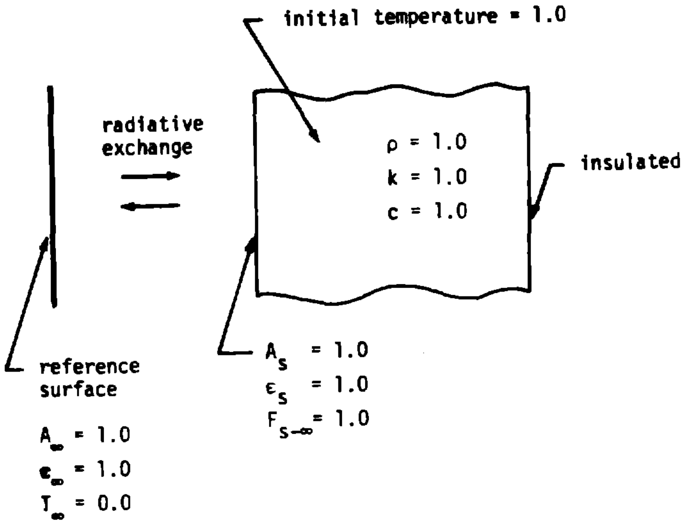


Figure 8.3a - Geometry and material properties for sample problem 3 - a one-material body with an insulated boundary and radiative exchange at the other boundary.

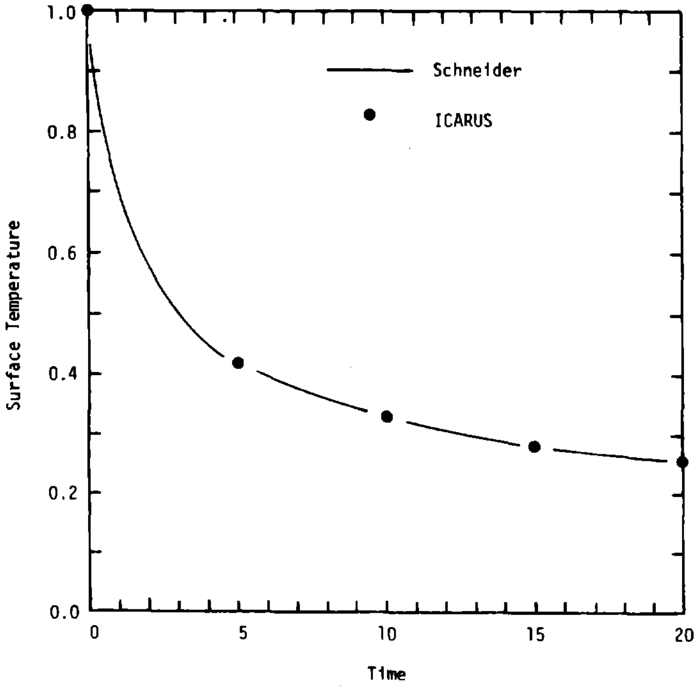


Figure 8.3b - Comparison of sample problem 3 results calculated by ICARUS with the analytical solution taken from Chart 52 of Schneider

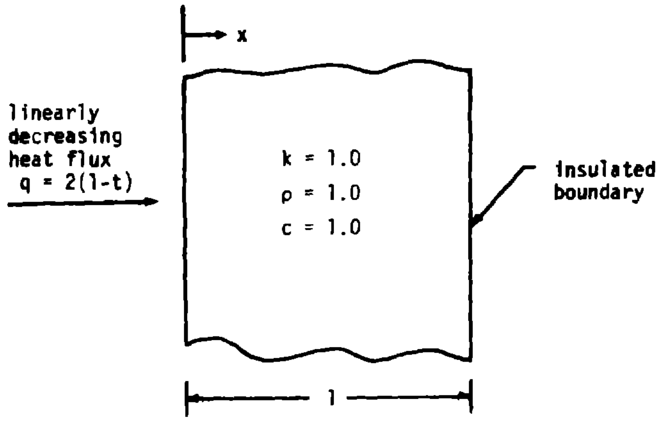


Figure 8.4a - Geometry and material properteis for sample problem 4 - a one-material body with an insulated surface and a heat flux applied at the other surface that decreases linearly with time.

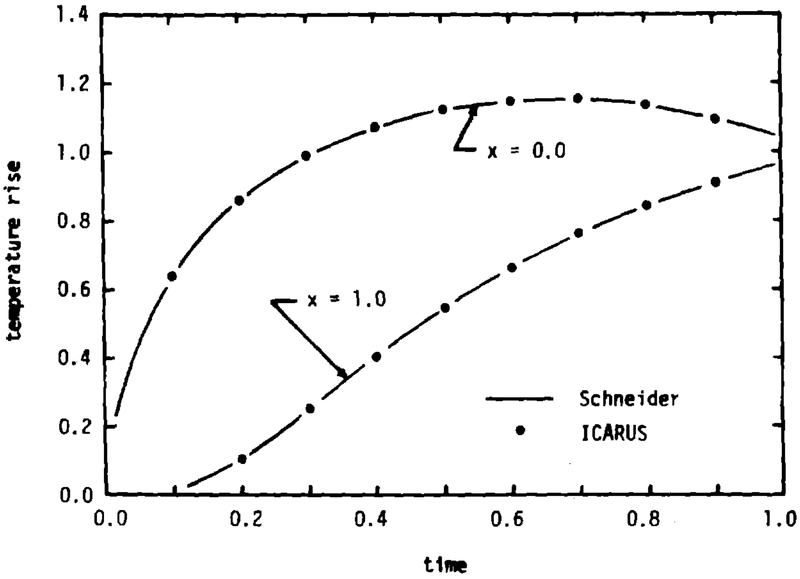


Figure 8.4b - Comparison of sample problem 4 results calculated by ICARUS with the analytical solution taken from Chart 46 of Schneider

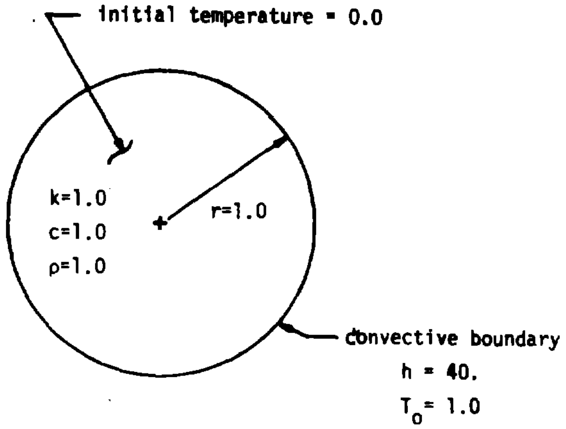


Figure 8.5a – Geometry and material properties for sample problem 5 – a solid sphere with convective cooling on the surface.

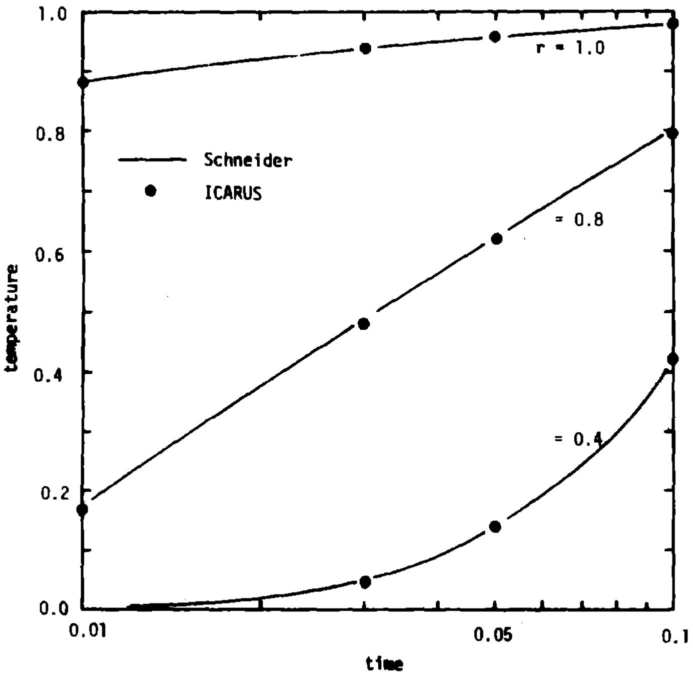


Figure 8.5b – Comparison of sample problem 5 results calculated by ICARUS with the analytical solution taken from Chart 37 of Schneider

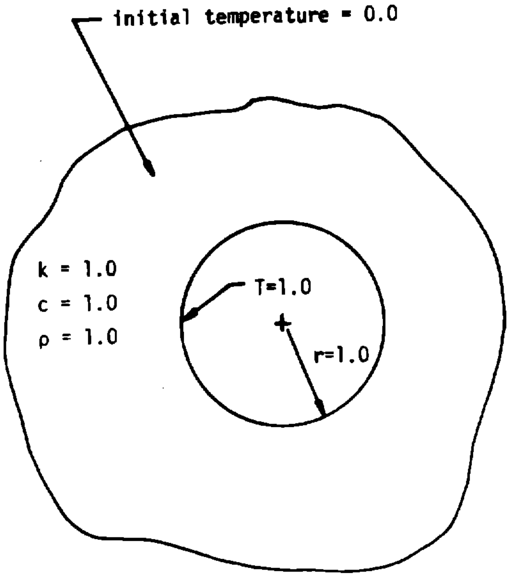


Figure 8.6a - Geometry and material properties for sample problem 6 - a infinite solid surrounding a spherical cavity with an applied temperature on the cavity boundary.

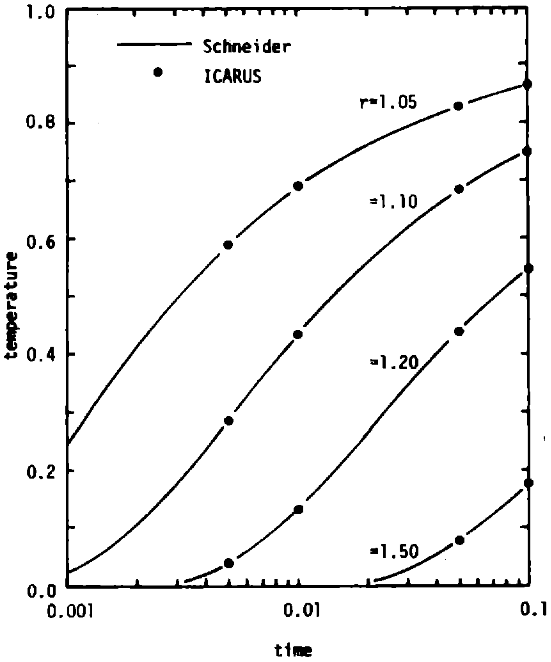


Figure 8.6b - Comparison of sample problem 6 results calculated by ICARUS with the analytical solution taken from Chart 9 of Schneider

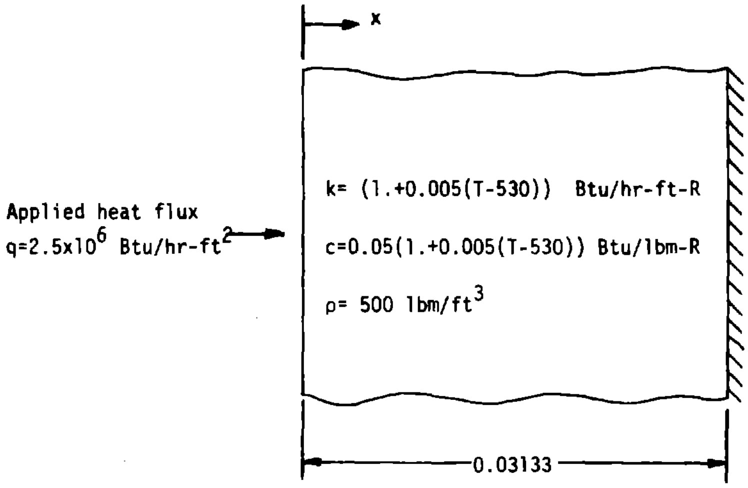

x
Applied heat flux
q=2.5x10^6 Btu/hr-ft^2
k= (1.+0.005(T-530)) Btu/hr-ft-R
c=0.05(1.+0.005(T-530)) Btu/lbm-R
ρ= 500 lbm/ft^3
0.03133

Figure 8.7 - Geometry and material properties for sample problem 7 - a single material body with an insulated boundary, applied heat flux boundary, and non-linear material properties.

Table 8.1

Temperature at x=0.

Time	Analytical	TOPAZ [7]	ICARUS
$1.\times10^{-6}$	847.19	845.44	844.37
$1.\times10^{-5}$	1201.41	1200.24	1199.3

9.0 ACKNOWLEDGMENTS

This work was performed under the auspices of the U. S. Department of energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48. The author is grateful to the Laboratories Micro-CAE program, under the leadership of Dr. William J. Comfort III, for the support necessary to develop this computer package.

10.0 REFERENCES

1. S. B. Sutton, ICARUS - A General One-Dimensional Heat Conduction Code, UCID-20125, Lawrence Livermore National Laboratory, July 20, 1984.

2. P. J. Roach, Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, 1972.

3. Microsoft Corporation, Microsoft FORTRAN Compiler for the MS-DOS Operating System - User's Guide - Version 3.2, Microsoft Corporation, 1984.

4. D. W. Stillman and D. B. Fields, P-PLOT User Manual, private communication, 1985.

5. International Business Machines Corp., BASIC Compiler by Microsoft - Version 1.0, IBM Corp., 1982.

6. P. J. Schneider, Temperature Response Charts, John Wiley, New York, 1963.

7. A. B. Shapiro, TOPAZ - A Finite Element Heat Conduction Code for Analyzing 2-D Solids, UCID-20045, Lawrence Livermore National Laboratory, March 1984.

APPENDIX A

Comments on Code Application

Zoning

- The accuracy of the solution is generally directly related to the mesh used. More accurate predictions will result from a finer mesh. This is particularly true near boundaries in very high applied flux situations. In these situations, the user must pay particular attention to the zoning near the boundaries to make sure that it is sufficient to capture the true temperature gradient.

- Zoning grading factors should generally not exceed 1.1 or be less that 0.9 (less than 10 percent change in zone width with each cell). Above these limits the spatial accuracy of the finite difference representations is diminished.

- A region must consist of no fewer than 2 zones.

Timestep

- The best timestep for a problem is very dependent on the characteristics of that problem. For accuracy, the best timestep is that near the explicit limit. As a rule-of-thumb, timesteps larger than 100 times the explicit limit are not advised. If fast boundary transients are present, the timestep must be maintained small enough to capture the time characterists of that condition.

Miscellaneous

- When using the pre-processors, if zoning is adjusted in the modify mode, be sure to also adjust the initial temperature specification.

- Before proceeding with input generation, a picture of the problem, including geometry, number of zones, interface node numbers, and material properties should be sketched. This picture will greatly aid in making input decisions and responses.

Heat flux sign convention

- On input, an applied boundary flux into the body is taken as positive.

- In the plot post-processing, a positive flux denotes left to right heat flow. Thus a flux into the body at the right boundary will be plotted as negative.

Non-linear iterations

- The iteration method is rather crude, so selection of the proper iteration relaxation factor is important. Under-relaxation or straight relaxation should always be used (the relaxation factor less than or equal to 1.0). Selection of proper relaxation paramaters (relaxation factor and convergence tolerence) is a trial-and-error proceedure that is problem dependent.

- Selcection of the relaxation factor is more important in steady-state problems than in transient problems. Many transient problems with strong material and boundary non-linearities have been calculated using a relaxation factor of 1.0. Steady-state calculations with strong material non-linearities have required a relaxation factor as small as 0.5.

Piecewise linear curves

- In ICARUSF, no extrapolation is done beyond the end points of the piecewise linear curve. Beyond the end points, the end point value is applied.

APPENDIX B

Definition of the Major Variables in the Heat Conduction Program ICARUSB

Variable Description

a(i) - the superdiagonal coefficient for the equation at nodal point i

ain(i) - the area of the lower index side of the control volume about nodal point i

aout(i) - the area of the upper index side of the control volume about nodal point i

ar1 - the reference radiation surface area associated with the lower index boundary

arm - the reference radiation surface area associated with the upper index boundary

b(i) - the diagonal coefficient for the equation at nodal point i

c(i) - the subdiagonal coefficient for the equation at nodal point i

cpm(k) - specific heat of material k

d(i) - the term containing the source terms for the equation at nodal point i

diffo(i) - coefficient at node i for diffusion contribution associated with node i

diffm(i) - coefficient at node i for diffusion contribution associated with node i-1

diffp(i) - coefficient at node i for diffusion contribution associated with node i+1

dr(i) - difference between coordinate value of node i and node i+1

dt - the problem timestep

ekm(k) - thermal conductivity of material k

eps - the convergence factor for non-linear iterations, defined as the ratio of the maximum change in temperature to the new temperature epsl(j) - emissivity on the left side of the gap between region j and region j+1

epsr(j) - emissivity on the right side of the gap between region j and region j+1

forml - the radiation form factor for the lower index boundary

formm - the radiation form factor for the upper index boundary

gforml(j) - gray body form factor for radiation transport from the left to right sides of the gap between region j and region j+1

gformr(j) - gray body form factor for radiation transport from the right to left sides of the gap between region j and region j+1

ghconl(j) - convection coefficient on the left side of the gap between region j and region j+1

ghconr(j) - convection coefficient on the right side of the gap between region j and region j+1

hl - the convective film coefficient at the lower index boundary

hm - the convective film coefficient at the upper index boundary

igeom - the geometry indicator (=1, planar; =2, cylindrical; =3, spherical)

ih(j) - upper nodal point value for region j

imatf(k) - flag to determine if material number k has been referenced (=1, material referenced, =0, material not referenced)

is(j) - lower nodal point value for region j

isteady - a control flag to specify if the problem is transient or steady-state (=0, steady-state; =1, transient)

iter - a control flag to specify if non-linear interation is to take place (=0, do not iterate; =1, iterate)

ith(j) - the upper index of temperature specification region j

itl(j) - the lower index of temperature specification region j

itmax - the maximum number of non-linar iterations allowed per time cycle itype(i) - model type for node i (=1, interior node not on interface, =2, node on material interface, =3, node at left edge of gap, =4, node at right edge of gap)

lb1 - the boundary type for the lower index boundary (=1, constant temperature; =2, general boundary condition)

lbm - the boundary type for the upper index boundary (=1, constant temperature; =2, general boundary condition)

matn(i) - material number at node i

matnam$(j) - name of material number j

matnum(j) - material number of region j

ncycle - the cycle counter

ngap - the number of material gaps present in the problem

npoint - the number of points to be declared for time history tracking

nstore - the cycle frequency for writing time history informtion to the time history plot dump file

ntemp - the number of temperature specification regions in the problem

nummat - the number of different materials in the problem

numreg - the number of geometric/material regions in the problem

qgen(j) - the internal heat generation for material number j

qr1 - the applied heat flux at the lower index boundary

qrm - the applied heat flux at the upper index boundary

r(i) - coordinate value at node i

relax - non-linear iteration relaxation factor

rh(j) - upper coordinate value of region j

rky(j) - geometric grading factor for region j

ro(j) - the density of material number j

rs(j) - lower coordinate value of region j sigma - the Stefan-Boltzmann constant

t(i) - current temperature at node i

tb1 - the specified temperature for the lower index boundary for the type 1 boundary condition

tbm - the specified temperature for the upper index boundary for the type 1 boundary condition

tedit1 - the time frequency for complete edits

th(j) - the temperature at the upper nodal point of temperature specification region j

theta - specifies the solution type (=0, explicit; =1, implicit)

time - the current problem time

tinf1 - the temperature of the reference radiation surface associated with the lower index boundary

tinfm - the temperature of the reference radiation surface associated with the upper index boundary

tl(j) - the temperature at the lower nodal point of temperature specification region j

tmax - the maximum problem time

told(i) - temporary intermediate value of temperature at nod i

tn(i) - old time level temperature at node i

tzero1 - the bulk fluid temperature for convection at the lower index boundary

tzerom - the bulk fluid temperature for convection at the upper index boundary

vbar(i) - volume of the control volume at node i

vqbar(i) - volumetric heat generation at node i

volm(i) - volume of the lower index half of the control volume at node i

volp(i) - volume of the upper index half of the control volume at node i

APPENDIX C

Definition of Major Variables in the Heat Conduction Program ICARUSF

Variable — Description

a(i) - the superdiagonal coefficient for the equation at nodal point i

ain(i) - the area of the lower index side of the control volume about nodal point i

aname - the problem title

aout(i) - the area of the upper index side of the control volume about nodal point i

ar1 - the reference radiation surface area associated with the lower index boundary

arm - the reference radiation surface area associated with the upper index boundary

b(i) - the diagonal coefficient for the equation at nodal point i

c(i) - the subdiagonal coefficient for the equation at nodal point i

ccmult(k) - the multiplier of the curve specified for the specific heat of material k

cmepsl(j) - the multiplier of the curve specifying the emissivity of the lower index boundary of gap number j

cmepsr(j) - the multiplier of the curve specifying the emissivity of the upper index boundary of gap number j

cname(m) - name of curve number m

ckmult(k) - the multiplier of the curve specified for the specific heat of material k

cmult1(k) - the curve multiplier for boundary condition property k at the lower index boundary

cmultm(k) - the curve multiplier for boundary condition property k at the lower index boundary

cp(i) - the calculated specific heat at nodal point i cpl(j) - the calculated specific heat at the lower index edge of region j

cpr(j) - the calculated specific heat at the upper index edge of region j

cqmult(k) - the multiplier of the curve specified for the temperature dependent heat generation of material k

cqtmult(k) - the multiplier of the curve specified for the time dependent heat generation of material k

d(i) - the term containing the source terms for the equation at nodal point i

dr(i) - difference between coordinate value of node i and node i+1

dt - the problem timestep

dte - the calculated explicit timestep

dtfac - timestep factor for implicit calculations (>0, specifies the implicit timestep; <0, the timestep used is the multiplier times the explicit timestep)

eps - the convergence factor for non-linear iterations, defined as the ratio of the maximum change in temperature to the new temperature

form1 - the radiation form factor for the lower index boundary

forml - gray body form factor at the left boundary

formm - the radiation form factor for the upper index boundary

formr - gray body form factor at the right boundary

gepsl(j) - emissivity on the left side of the gap between region j and region j+1

gepsr(j) - emissivity on the right side of the gap between region j and region j+1

gforml(j) - gray body form factor for radiation transport from the left to right sides of the gap between region j and region j+1

gformr(j) - gray body form factor for radiation transport from the right to left sides of the gap between region j and region j+1

ghconl(j) - convection coefficient on the left side of the gap between region j and region j+1 ghconr(j) - convection coefficient on the right side of the gap between region j and region j+1

h1 - convection film coefficient at the left boundary

hm - convection film coefficient at the right boundary

iedit1 - specifies the long edit type (=0, compact; =1, extensive)

igeom - the geometry indicator (=1, planar; =2, cylindrical; =3, spherical)

ih(j) - upper nodal point value for region j

is(j) - lower nodal point vale for region j

isteady - a control flag to specify if the problem is transient or steady-state (=0, steady-state; =1, transient)

iter - a control flag to specify if non-linear interation is to take place (=0, do not iterate; =1, iterate)

ith(j) - the upper index of temperature specification region j

itl(j) - the lower index of temperature specification region j

itmax - the maximum number of non-linar iterations allowed per time cycle

itype(i) - model type for node i (=1, interior node not on interface, =2, node on material interface, =3, node at left edge of gap, =4, node at right edge of gap)

lb1 - the boundary type for the lower index boundary (=1, constant temperature; =2, general boundary condition)

lbm - the boundary type for the upper index boundary (=1, constant temperature; =2, general boundary condition)

matn(i) - material number at node i

matnam(j) - name of material number j

matnum(j) - material number of region j

mup - the value of the index at the upper index boundary

nbc(k) - the number of entries in piecewise linear curve k

nc(j) - the curve number for the specific heat of material j ncepsl(j) - the curve number for the emissivity at the lower index edge of the gap between regions j and j+1

ncepsr(j) - the curve number for the emissivity at the upper index edge of the gap between regions j and j+1

ncurvl(k) - the curve number for boundary property k at the lower index boundary

ncurvm(k) - the curve number for boundary property k at the upper index boundary

ncycle - the cycle counter

neditl - cycle frequency for long edits

nedits - cycle frequency for short edits

ngap - the number of material gaps present in the problem

nk(j) - the curve number for the thermal conductivity of material number j

npoint - the number of points to be declared for time history tracking

npts(j) - the points for time history storage

nq(j) - the curve number for the temperature dependent heat generation for material number j

nqt(j) - the curve number for the time dependent heat generation for material number j

nstore - the cycle frequency for writing time history informtion to the time history plot dump file

ntemp - the number of temperature specification regions in the problem

nummat - the number of different materials in the problem

numreg - the number of geometric/material regions in the problem

qn(i) - the total internal heat generation at nodal point i

qrl - applied heat flux at the left boundary

qrm - applied heat flux at the right boundary

r(i) - coordinate value at node i

relax - non-linear iteration relaxation factor rh(j) - upper coordinate value of region j

rky(j) - geometric grading factor for region j

ro(j) - the density of material number j

rs(j) - lower coordinate value of region j

sigma - the Stefan-Boltzmann constant

t(i) - current temperature at node i

tcurv(l,m) - the time/temperature entries for curve number m

teditl - the time frequency for complete edits

th(j) - the temperature at the upper nodal point of temperautre specification region j

theta - specifies the solution type (=0, explicit; =1, implicit)

time - the current problem time

tinfl - reference radiation temperature at the left boundary

tinfm - reference radiation temperature at the right boundary

tl(j) - the temperature at the lower nodal point of temperature specification region j

tmax - the maximum problem time

tol - reference convection fluid bulk temperature for the left boundary

told(i) - temporary intermediate value of temperature at nod i

tom - reference convection fluid bulk temperature for the right boundary

tn(i) - old time level temperature at node i

vcurv(l,m) - the curve values for curve number m

volm(i) - volume of the lower index half of the control volume at node i

volp(i) - volume of the upper index half of the control volume at node i

APPENDIX D

Definition of Major Variables in the Pre-Processor Program PREICAR

Variable	Description
areal	- the reference radiation surface area associated with the lower index boundary
aream	- the reference radiation surface area associated with the upper index boundary
ccmult(k)	- the multiplier of the curve specified for the specific heat of material k
cmepsl(j)	- the multiplier of the curve specifying the emissivity of the lower index boundary of gap number j
cmepsr(j)	- the multiplier of the curve specifying the emissivity of the upper index boundary of gap number j
ckmult(k)	- the multiplier of the curve specified for the specific heat of material k
cmult1(k)	- the curve multiplier for boundary condition property k at the lower index boundary
cmultm(k)	- the curve multiplier for boundary condition property k at the lower index boundary
cqmult(k)	- the multiplier of the curve specified for the temperature dependent heat generation of material k
cqtmult(k)	- the multiplier of the curve specified for the time dependent heat generation of material k
curvnam$(m)	- name of curve number m
dt	- the problem timestep
dtfac	- timestep multiplication factor for implicit calculations (>0, specifies the implicit timestep; <0, the multiplier times the explicit timestep)
eps	- the convergence factor for non-linear iterations, defined as the ratio of the maximum change in temperature to the new temperature
forml	- the radiation form factor for the lower index boundary
formm	- the radiation form factor for the upper index boundary

ghconl(j) - convection coefficient on the left side of the gap between region j and region j+1

ghconr(j) - convection coefficient on the right side of the gap between region j and region j+1

iedit1 - specifies the long edit type (=0, compact; =1, extensive)

igeom - the geometry indicator (=1, planar; =2, cylindrical; =3, spherical)

ih(j) - upper nodal point value for region j

is(j) - lower nodal point vale for region j

isteady - a control flag to specify if the problem is transient or steady-state (=0, steady-state; =1, transient)

iter - a control flag to specify if non-linear interation is to take place (=0, do not iterate; =1, iterate)

ith(j) - the upper index of temperature specification region j

itl(j) - the lower index of temperature specification region j

itmax - the maximum number of non-linar iterations allowed per time cycle

lb1 - the boundary type for the lower index boundary (=1, constant temperature; =2, general boundary condition)

lbm - the boundary type for the upper index boundary (=1, constant temperature; =2, general boundary condition)

matnam$(k) - name of material number k

matnum(j) - material number of region j

nbc(k) - the number of entries in piecewise linear curve k

nc(j) - the curve number for the specific heat of material j

ncepsl(j) - the curve number for the emissivity at the lower index edge of the gap between regions j and j+1

ncepsr(j) - the curve number for the emissivity at the upper index edge of the gap between regions j and j+1

ncurv1(k) - the curve number for boundary property k at the lower index boundary ncurvm(k) - the curve number for boundary property k at the upper index boundary

neditl - cycle frequency for long edits

nedits - cycle frequency for short edits

ngap - the number of material gaps present in the problem

nk(j) - the curve number for the thermal conductivity of material number j

npoint - the number of points to be declared for time history tracking

npts(j) - the points for time history storage

nq(j) - the curve number for the temperature dependent heat generation for material number j

nqt(j) - the curve number for the time dependent heat generation for material number j

nstore - the cycle frequency for writing time history informtion to the time history plot dump file

ntemp - the number of temperature specification regions in the problem

nummat - the number of different materials in the problem

numreg - the number of geometric/material regions in the problem

relax - non-linear iteration relaxation factor

rh(j) - upper coordinate value of region j

rky(j) - geometric grading factor for region j

ro(j) - the density of material number j

rs(j) - lower coordinate value of region j

sigma - the Stefan-Boltzmann constant

tcurv(l,m) - the time/temperature entries for curve number m

teditl - the time frequency for complete edits

th(j) - the temperature at the upper nodal point of temperautre specification region j theta - specifies the solution type (=0, explicit; =1, implicit)

tl(j) - the temperature at the lower nodal point of temperature specification region j

tmax - the maximum problem time

vcurv(l,m) - the curve values for curve number m

APPENDIX E

Definition of Major Variables in the Plot Post-processing Programs PLOTIC and CPLOTIC

Variable	Description
aname$	- problem title
cxmax	- the horizontal raster count
cymax	- the vertical raster count
gxmin	- the minimum horizontal screen limit for the plot grid
gxmax	- the maximum horizontal screen limit for the plot grid
gymin	- the minimum vertical screen limit for the plot grid
gymax	- the maximum vertical screen limit for the plot grid
igeom	- geometry indicator (=1, planar; =2, cylindrical; =3, spherical)
ih(j)	- upper nodal point value for region j
is(j)	- lower nodal point value for region j
isnap$	- file descriptor for the snapshot plot data file
isteady	- specifies if the ICARUS problem was transient or steady-state (=0, transient; =1, steady-state)
itime$	- file descriptor for the time history plot data file
maxtim	- maximum plot time for the time history plot
mintim	- minimum plot time for the time history plot
mup	- total number of nodal points in the problem
nend	- maximum node number for the spatial plot or grid plot
npoint	- number of nodal points for which time history data was stored during the heat conduction solution
npts(k)	- the nodal points having data stored for time history plotting
nscyc(k)	- the calculation cycle of snapshot plot dump number k

nsdump - the number of snapshot plot dump states in the data file

nstart - minimum node number for the spatial plot or grid plot

numnx - number of coordinate numbers on abscissa

numny - number of coordinate numbers on ordinate

numpnt - number of entries in the time history data for each point

numreg - number of material/geometry regions in the problem

numtx - number of tick marks on abscissa

numty - number of tick marks on ordinate

powx - the scale power factor for the abscissa

powy - the scale power factor for the ordinate

r(i) - the coordinate value at nodal point k

rh(k) - lower coordinate value of region k

rky(k) - geometric grading factor for region k

rs(k) - upper coordinate value of region k

rval(k) - the coordinate value of the nodal points having data stored for time history plotting

stime(k) - the calculation time of snapshot plot dump number k

tv(i) - the temperature or heat flux at nodal point i

tempv(k) - the temperature or heat flux at entry number k in the time history data block

time(k) - the time at entry point k in the time history data block

tmax - the maximum temperature in the data for the plot

tmin - the minimum temperature in the data for the plot